

RP04,05,06

DUAL CONTROLLER LOGIC
CZRJECO

AH-9201C-MC

COPYRIGHT ©76-78

FICHE 1 OF 1

APR 1978

digital

MADE IN USA

The image displays a large grid of technical data, organized into 15 columns and 20 rows. Each cell within the grid contains small, dense text and symbols, characteristic of a truth table or logic diagram. The text is too small to be legible, but the overall structure suggests a systematic arrangement of binary values and logic symbols. The grid is set against a dark background, and the text is light-colored, creating a high-contrast visual. The layout is consistent across the entire page, with no significant variations in cell content or structure.

5570
5571
5572
5573
5574
5575
5576
5577
5578
5579
5580
5581
5582
5583
5584
5585
5586
5587

.REM 2

IDENTIFICATION

PRODUCT CODE: AC-9200C-MC
PRODUCT NAME: CZRJECO RPO4/5/6 DL CTRLR LGC
DATE CREATED: MARCH 1978
MAINTAINER: DIAGNOSTIC ENGINEERING
AUTHOR: C. HESS

THE INFORMATION IN THIS DOCUMENT IS SUBJECT TO CHANGE WITHOUT NOTICE AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT CORPORATION. DIGITAL EQUIPMENT CORPORATION ASSUMES NO RESPONSIBILITY FOR ANY ERRORS THAT MAY APPEAR IN THIS DOCUMENT.

NO RESPONSIBILITY IS ASSUMED FOR THE USE OR RELIABILITY OF SOFTWARE ON EQUIPMENT THAT IS NOT SUPPLIED BY DIGITAL OR ITS AFFILIATED COMPANIES.

COPYRIGHT (C) 1976,1978 BY DIGITAL EQUIPMENT CORPORATION

THE FOLLOWING ARE TRADEMARKS OF DIGITAL EQUIPMENT CORPORATION:

DIGITAL PDP UNIBUS MASSBLS
DEC DECUS DECTAPE

5589
5590
5591
5592
5593
5594
5595
5596
5597
5598
5599
5600
5601
5602
5603
5604
5605
5606
5607
5608
5609
5610
5611
5612
5613
5614
5615
5616
5617
5618
5619
5620
5621
5622
5623
5624
5625
5626
5627
5628
5629
5630
5631
5632
5633

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

5635
5636
5637
5638
5639
5640
5641
5642
5643
5644
5645
5646
5647
5648
5649
5650
5651
5652
5653
5654
5655
5656
5657
5658
5659
5660
5661
5662
5663
5664
5665
5666
5667
5668
5669
5670
5671
5672
5673
5674
5675
5676
5677
5678
5679
5680
5681
5682
5683
5684
5685
5686
5687
5688
5689
5690

1. ABSTRACT

 THE RPO4/5/6 DUAL CONTROLLER LOGIC TEST PERFORMS A SERIES OF TESTS WHICH VERIFY THAT THE RPO4/5/6 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 RPO4/5/6
- RPO4/5/6 DUAL CONTROLLER OPTION TEST CABLE

2.2 PRELIMINARY PROGRAMS

RPO4/5/6 DISKLESS CONTROLLER TEST
 PART 1 (MAINDEC-11-DZRJG)
 PART 2 (MAINDEC-11-DZRJH)

RPO4/5/6 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 RPO4/5/6 DUAL CONTROLLER LOGIC TEST, PART 2 (MAINDEC-11-DZRJF).
- B. DYNAMIC OPERATION OF THE DUAL CONTROLLER OPTION IS TESTED BY THE RPO4/5/6 MULTIDRIVE EXERCISER PROGRAM (MAINDEC-11-DZRJD).

5691
5692
5693
5694
5695
5696
5697
5698
5699
5700
5701
5702
5703
5704
5705
5706
5707
5708
5709
5710
5711
5712
5713
5714
5715
5716
5717
5718
5719
5720
5721
5722
5723
5724
5725
5726
5727
5728
5729
5730
5731
5732
5733
5734
5735
5736
5737
5738
5739
5740
5741
5742
5743
5744
5745
5746

3. LOADING PROCEDURES

 THE PROGRAM MAY BE LOADED BY THE ABSOLUTE PAPER TAPE LOADER OR IT MAY BE LOADED FROM THE APPROPRIATE MEDIA USING THE ASSOCIATED 'XXDP' 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 APPROPRIATE 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 RPO4, OR THE NUMBER OF THE ADDRESS PLUG IF THE DRIVE IS AN RPO5/6).
 - 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 RPO4/S/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.

5747
5748
5749
5750
5751
5752
5753
5754
5755
5756
5757
5758
5759
5760
5761
5762
5763
5764
5765
5766
5767
5768
5769
5770
5771
5772
5773
5774
5775
5776
5777
5778
5779
5780
5781
5782
5783
5784
5785
5786
5787
5788
5789
5790
5791
5792
5793
5794
5795
5796
5797
5798
5799
5800
5801
5802

5803
5804
5805
5806
5807
5808
5809
5810
5811
5812
5813
5814
5815
5816
5817
5818
5819
5820
5821
5822
5823
5824
5825
5826
5827
5828
5829
5830
5831
5832
5833
5834
5835
5836
5837
5838
5839
5840
5841
5842
5843
5844
5845
5846
5847
5848
5849
5850
5851
5852
5853
5854
5855
5856
5857
5858

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 RPO4/5/6 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 RPO4 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 RPO4'S, OR BY THE ADDRESS PLUG FOR RPO5/6'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 CONNFCTED, THE ATTENTION

5859
5860
5861
5862
5863
5864
5865
5866
5867
5868
5869
5870
5871
5872
5873
5874
5875
5876
5877
5878
5879
5880
5881
5882
5883
5884
5885
5886
5887
5888
5889
5890
5891
5892
5893
5894
5895
5896
5897
5898
5899
5900
5901
5902
5903
5904
5905
5906
5907
5908
5909
5910
5911
5912
5913
5914

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 APPROPRIATE REGISTERS

7. MISCELLANEOUS

7.1 RESTRICTIONS

TO RUN THIS PROGRAM, THE SYSTEM MUST HAVE EITHER A KW11-P OR A KW11-L CLOCK. ADDITIONALLY, THE 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.

915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970

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 RO-R5
- RESREG - ROUTINE TO RESTORE RO-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

5971
5972
5973
5974
5975
5976
5977
5978
5979
5980
5981
5982
5983
5984
5985
5986
5987
5988
5989
5990
5991
5992
5993
5994
5995
5996
5997
5998
5999
6000
6001
6002
6003
6004
6005
6006
6007
6008
6009
6010
6011
6012
6013
6014
6015
6016
6017
6018
6019
6020
6021
6022
6023
6024
6025
6026

TEST NO. (IN DECIMAL)

01	1
02	10
03	10
04	1
05	1
06	400
07	4000
10	100
11	100
12	400
13	f
14	f
15	f
16	400
17	400
20	400
21	400
22	400
23	400
24	400
25	400
26	400
27	400
30	400
31	400
32	400
33	400
34	400
35	400
36	400
37	400
40	400
41	400
42	400
43	400
44	400
45	400
46	20
47	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 RPO4 TO A KNOWN STATE BEFORE LOOPING ON THE ERROR, THE TEST FOR SW<09> IS PERFORMED AT THE END OF THE TEST - NOT AT THE POINT WHERE THE ERROR

6027
6028
6029
6030
6031
6032
6033
6034
6035
6036
6037
6038
6039
6040
6041
6042
6043
6044
6045
6046
6047
6048
6049
6050
6051
6052
6053
6054
6055
6056
6057
6058
6059
6060
6061
6062
6063
6064
6065
6066
6067
6068
6069
6070
6071
6072
6073
6074
6075
6076
6077
6078
6079
6080
6081
6082

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 DESCREPCANCY 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 VERIFYING 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 RPO4/5/6, 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.

6083
6084
6085
6086
6087
6088
6089
6090
6091
6092
6093
6094
6095
6096
6097
6098
6099
6100
6101
6102
6103
6104
6105
6106
6107
6108
6109
6110
6111
6112
6113
6114
6115
6116
6117
6118
6119
6120
6121
6122
6123
6124
6125
6126
6127
6128
6129
6130
6131
6132
6133
6134
6135
6136
6137
6138

- 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 RPCS1 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 TIMEOUT ONE SHOT AND SAVE THE
VALUE FOR LATER USE. VERIFY THAT TIMEOUT RETURNED THE DRIVE TO
NEUTRAL.
- 8.6 TEST 4 - PORT 'A' COMMAND SEIZE TEST & SET 'VV-A'
- VERIFY THAT THE DRIVE IS SEIZED WHEN A COMMAND IS ISSUED. SET 'VV'
FOR THE PORT UNDER TEST.
- A. ISSUE A DRIVE CLEAR COMMAND THROUGH PORT 'A'. VERIFY THAT THE
DRIVE WAS SEIZED BY PORT 'A' AND THAT THE 'GO' BIT RESET.
- B. ISSUE A READIN PRESET COMMAND THROUGH PORT 'A'. VERIFY THAT THE
'VV' BIT WAS SET FOR PORT 'A' AND THAT THE 'VV' BIT WAS NOT SET
FOR PORT 'B'. (NOTE THAT THE 'VV' BIT NOT BEING SET FOR PORT
'B' CAN ONLY BE TESTED THE FIRST TIME THROUGH THE PROGRAM.)
- C. STALL FOR 2 SECONDS THEN VERIFY THAT THE PORT TIMEOUT RELEASED
THE DRIVE AND THE THE DRIVE RETURNED TO NEUTRAL.
- 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 SEIZED BY PORT 'B' AND THAT THE 'GO' BIT RESET.
- B. ISSUE A READIN PRESET COMMAND THROUGH PORT 'B'. VERIFY THAT THE
'VV' BIT FOR PORT 'B' WAS SET.
- C. STALL FOR 2 SECONDS THEN VERIFY THAT THE PORT TIMEOUT RELEASED
THE DRIVE AND THE THE DRIVE RETURNED TO NEUTRAL.
- 8.8 TEST 6 - TEST RELEASE, DRIVE SEIZED BY PORT 'A'

6139
6140
6141
6142
6143
6144
6145
6146
6147
6148
6149
6150
6151
6152
6153
6154
6155
6156
6157
6158
6159
6160
6161
6162
6163
6164
6165
6166
6167
6168
6169
6170
6171
6172
6173
6174
6175
6176
6177
6178
6179
6180
6181
6182
6183
6184
6185
6186
6187
6188
6189
6190
6191
6192
6193
6194

- 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'

6195
6196
6197
6198
6199
6200
6201
6202
6203
6204
6205
6206
6207
6208
6209
6210
6211
6212
6213
6214
6215
6216
6217
6218
6219
6220
6221
6222
6223
6224
6225
6226
6227
6228
6229
6230
6231
6232
6233
6234
6235
6236
6237
6238
6239
6240
6241
6242
6243
6244
6245
6246
6247
6248
6249
6250

6251
6252
6253
6254
6255
6256
6257
6258
6259
6260
6261
6262
6263
6264
6265
6266
6267
6268
6269
6270
6271
6272
6273
6274
6275
6276
6277
6278
6279
6280
6281
6282
6283
6284
6285
6286
6287
6288
6289
6290
6291
6292
6293
6294
6295
6296
6297
6298
6299
6300
6301
6302
6303
6304
6305
6306

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

6307
6308
6309
6310
6311
6312
6313
6314
6315
6316
6317
6318
6319
6320
6321
6322
6323
6324
6325
6326
6327
6328
6329
6330
6331
6332
6333
6334
6335
6336
6337
6338
6339
6340
6341
6342
6343
6344
6345
6346
6347
6348
6349
6350
6351
6352
6353
6354
6355
6356
6357
6358
6359
6360
6361
6362

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 RPS1.

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 RPS1.

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'.

- 6363
6364
6365
6366
6367
6368
6369
6370
6371
6372
6373
6374
6375
6376
6377
6378
6379
6380
6381
6382
6383
6384
6385
6386
6387
6388
6389
6390
6391
6392
6393
6394
6395
6396
6397
6398
6399
6400
6401
6402
6403
6404
6405
6406
6407
6408
6409
6410
6411
6412
6413
6414
6415
6416
6417
6418
- E. RELEASE THE DRIVE THROUGH PORT 'B'. VERIFY THAT THE DRIVE RETURNED TO NEUTRAL AND THAT NEITHER ATTENTION BIT IS SET.
- 8.25 TEST 27 - TEST SEIZE BY WRITING ATTENTION BIT
TEST THAT WRITING THE APPROPRIATE DRIVE BIT INTO THE ATTENTION REGISTER (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, INTO THE ATTENTION REGISTER.
- B. VERIFY THAT THE DRIVE IS STILL IN NEUTRAL.
- 8.27 TEST 31 - TEST PORT 'A' TIMEOUT DOES NOT RESET DRIVE
VERIFY THAT PORT TIMEOUT DOES NOT INITIALIZE THE DRIVE.
- A. SEIZE THE DRIVE THROUGH PORT 'A' BY WRITING 0'S INTO 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 DRIVE TO TIMEOUT. VERIFY THAT THE DRIVE RETURNED TO NEUTRAL; THAT ATTENTION IS SET FOR PORT 'B' AND IS NOT SET FOR PORT 'A'; AND THAT BOTH PORTS SEE 1'S IN THE ERROR REGISTER.
- 8.29 TEST 33 - TEST RELEASE THROUGH PORT 'A' WITH ERRORS SET
VERIFY THAT A RELEASE COMMAND PERFORMS NO ACTION IF ISSUED WHEN ERROR BITS ARE SET IN THE DRIVE.
- A. SEIZE THE DRIVE THROUGH PORT 'A' BY WRITING 0'S INTO RPDS1.

6419
6420
6421
6422
6423
6424
6425
6426
6427
6428
6429
6430
6431
6432
6433
6434
6435
6436
6437
6438
6439
6440
6441
6442
6443
6444
6445
6446
6447
6448
6449
6450
6451
6452
6453
6454
6455
6456
6457
6458
6459
6460
6461
6462
6463
6464
6465
6466
6467
6468
6469
6470
6471
6472
6473
6474

- B. WRITE 1'S INTO RPER1 THROUGH PORT 'A'.
- C. ISSUE A RELEASE COMMAND THROUGH PORT 'A'. VERIFY THAT THE 'GO' BIT HAS RESET, THAT THE DRIVE HAS NOT RETURNED TO NEUTRAL, AND THAT RPER1 HAS NOT BEEN CLEARED.
- D. CLEAR RPER1 BY ISSUING A DRIVE CLEAR COMMAND THROUGH PORT 'A'.
- E. ISSUE A RELEASE COMMAND THROUGH PORT 'A'. VERIFY THAT THE DRIVE RETURNED TO NEUTRAL AND THAT NEITHER ATTENTION BIT IS SET.

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

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

- A. SEIZE THE DRIVE THROUGH PORT 'B' BY WRITING 0'S INTO RPDS1.
- B. WRITE 1'S INTO RPER1 THROUGH PORT 'B'.
- C. ISSUE A RELEASE COMMAND THROUGH PORT 'B'. VERIFY THAT THE 'GO' BIT HAS RESET, THAT THE DRIVE HAS NOT RETURNED TO NEUTRAL, AND THAT RPER1 HAS NOT BEEN CLEARED.
- D. CLEAR RPER1 BY ISSUING A DRIVE CLEAR COMMAND THROUGH PORT 'B'.
- E. ISSUE A RELEASE COMMAND THROUGH PORT 'B'. VERIFY THAT THE DRIVE RETURNED TO NEUTRAL AND THAT NEITHER ATTENTION BIT IS SET.

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

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

- A. SEIZE THE DRIVE THROUGH PORT 'A' BY WRITING 0'S INTO RPDS1.
- B. WAIT 500 MS AND WRITE 0'S INTO RPDS1 THROUGH PORT 'A'.
- C. VERIFY THAT THE TIMEOUT OCCURS 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.

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

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

- A. ISSUE A RECALIBRATE COMMAND THROUGH PORT 'A'.
- B. WAIT FOR THE RECALIBRATE COMMAND TO COMPLETE ('DRY' TO BECOME '1'). VERIFY THAT THE ATTENTION BIT FOR PORT 'A' IS SET AND THAT THE ATTENTION BIT FOR PORT 'B' IS NOT SET.
- C. RELEASE THE DRIVE THROUGH PORT 'A'. VERIFY THAT THE DRIVE RETURNED

TO NEUTRAL AND THAT NEITHER ATTENTION BIT IS SET.

- 8.34 TEST 40 - TEST PORT 'B' ATTENTION AFTER A COMMAND
TEST THE OPERATION OF THE PROT A AND PORT B ATTENTION BITS AFTER A COMMAND.
- A. ISSUE A RECALIBRATE COMMAND THROUGH PORT 'B'.
 - B. WAIT FOR THE RECALIBRATE COMMAND TO COMPLETE ('DRY' TO BECOME '1'). VERIFY THAT THE ATTENTION BIT FOR PORT 'B' IS SET AND THAT THE ATTENTION BIT FOR PORT 'A' IS NOT SET.
 - C. RELEASE THE DRIVE THROUGH PORT 'B'. VERIFY THAT THE DRIVE RETURNED TO NEUTRAL AND THAT NEITHER ATTENTION BIT IS SET.
- 8.35 TEST 41 - TEST PORT INTERACTION FROM PORT 'A'
VERIFY THAT THERE IS NO INTERACTION BETWEEN PORTS.
- A. SEIZE THE DRIVE THROUGH PORT 'A' BY WRITING 0'S INTO RPS1.
 - 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.
- 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 RPS1.
 - 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

6475
6476
6477
6478
6479
6480
6481
6482
6483
6484
6485
6486
6487
6488
6489
6490
6491
6492
6493
6494
6495
6496
6497
6498
6499
6500
6501
6502
6503
6504
6505
6506
6507
6508
6509
6510
6511
6512
6513
6514
6515
6516
6517
6518
6519
6520
6521
6522
6523
6524
6525
6526
6527
6528
6529
6530

6531 SWITCHED TO PORT 'A' AND THAT THE ATTENTION BIT FOR PORT 'A' IS
6532 SET AND THE ATTENTION BIT FOR PORT 'B' IS NOT SET.
6533
6534 G. ISSUE A RELEASE COMMAND THROUGH PORT 'A'. VERIFY THAT THE DRIVE
6535 RETURNED TO NEUTRAL AND THAT NEITHER ATTENTION BIT IS SET.
6536
6537 B.37 TEST 43 - TEST PORT 'A' ALTERNATE ATTENTION BIT PATH
6538 VERIFY THAT THE ALTERNATE ATTENTION REGISTER READ PATH IS OPERATIONAL.
6539
6540 A. SET THE ATTENTION BIT FOR PORT 'A'.
6541
6542 B. SEIZE THE DRIVE THROUGH PORT 'B' BY WRITING 0'S INTO RPDS1.
6543
6544 C. READ THE ATTENTION REGISTER & VERIFY THAT THE ATTENTION BIT
6545 FOR THE DRIVE IS SET.
6546
6547 B.38 TEST 44 - TEST PORT 'B' ALTERNATE ATTENTION BIT PATH
6548 VERIFY THAT THE ALTERNATE ATTENTION REGISTER READ PATH IS OPERATIONAL.
6549
6550 A. SET THE ATTENTION BIT FOR PORT 'B'.
6551
6552 B. SEIZE THE DRIVE THROUGH PORT 'A' BY WRITING 0'S INTO RPDS1.
6553
6554 C. READ THE ATTENTION REGISTER & VERIFY THAT THE ATTENTION BIT
6555 FOR THE DRIVE IS SET.
6556
6557 B.39 TEST 45 - TEST NO TIMEOUT THROUGH PORT 'A'
6558 VERIFY THAT THE TIMEOUT ONE-SHOT IS NOT TRIGGERED WHEN THE DRIVE
6559 SWITCHES PORTS AND SEIZING PORT PERFORMS NO REGISTER ACCESSES.
6560 THIS TEST IS FOR DRIVES WHICH HAVE THE M7775 'DP' BOARD AND IS
6561 NOT RUN AS PART THE TEST SEQUENCE. TO RUN THIS TEST, IT MUST
6562 BE SELECTED SEPARATELY.
6563
6564 A. SEIZE THE DRIVE THROUGH PORT 'B' BY WRITING 0'S INTO RPDS1.
6565
6566 B. SET PORT REQUEST BY WRITING 0'S INTO RPDS1 FROM PORT 'A'.
6567
6568 C. ISSUE A RELEASE COMMAND FROM PORT 'B'. VERIFY THAT THE DRIVE
6569 HAS SWITCHED TO THE OTHER PORT AND THAT THE 'ATA' BIT DID NOT
6570 SET FOR PORT 'B'. REGISTERS WILL NOT BE CHECKED THROUGH PORT 'A'.
6571
6572 D. WAIT THE TIMEOUT INTERVAL + 25%. VERIFY THAT THE DRIVE HAS NOT
6573 BEEN RELEASED.
6574
6575 E. RELEASE THE DRIVE THROUGH PORT 'A'. VERIFY THAT THE DRIVE
6576 RETURNED TO NEUTRAL AND THAT NEITHER ATTENTION BIT IS SET.
6577
6578 B.40 TEST 46 - TEST NO TIMEOUT THROUGH PORT 'B'
6579 VERIFY THAT THE TIMEOUT ONE-SHOT IS NOT TRIGGERED WHEN THE DRIVE
6580 SWITCHES PORTS AND SEIZING PORT PERFORMS NO REGISTER ACCESSES.
6581 THIS TEST IS FOR DRIVES WHICH HAVE THE M7775 'DP' BOARD AND IS
6582 NOT RUN AS PART THE TEST SEQUENCE. TO RUN THIS TEST, IT MUST
6583
6584
6585
6586

6587
6588
6589
6590
6591
6592
6593
6594
6595
6596
6597
6598
6599
6600
6601
6602
6603
6604
6605
6606
6607
6608
7141
7142
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
7143
7144
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
7145
7146
7147
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
000C11
000C12

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

2

```
.TITLE CZRJECO, 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 ***
STACK= 1100
.EQUIV EMT,ERROR ;;BASIC DEFINITION OF ERROR CALL
.EQUIV IOT,SCOPE ;;BASIC DEFINITION OF SCOPE CALL
*
*MISCELLANEOUS DEFINITIONS
HT= 11 ;;CODE FOR HORIZONTAL TAB
LF= 12 ;;CODE FOR LINE FEED
```

```

(1) 000015 CR= 15 :: CODE FOR CARRIAGE RETURN
(1) 000200 CRLF= 200 :: CODE FOR CARRIAGE RETURN-LINE FEED
(1) 177776 PS= 177776 :: PROCESSOR STATUS WORD
(1) .EQUIV PS,PSW
(1) 177774 STKLMT= 177774 :: STACK LIMIT REGISTER
(1) 177772 PIRQ= 177772 :: PROGRAM INTERRUPT REQUEST REGISTER
(1) 177570 DSWR= 177570 :: HARDWARE SWITCH REGISTER
(1) 177570 DDISP= 177570 :: HARDWARE DISPLAY REGISTER

```

.*GENERAL PURPOSE REGISTER DEFINITIONS

```

(1) 000000 R0= %0 :: GENERAL REGISTER
(1) 000001 R1= %1 :: GENERAL REGISTER
(1) 000002 R2= %2 :: GENERAL REGISTER
(1) 000003 R3= %3 :: GENERAL REGISTER
(1) 000004 R4= %4 :: GENERAL REGISTER
(1) 000005 R5= %5 :: GENERAL REGISTER
(1) 000006 R6= %6 :: GENERAL REGISTER
(1) 000007 R7= %7 :: GENERAL REGISTER
(1) 000006 SP= %6 :: STACK POINTER
(1) 000007 PC= %7 :: PROGRAM COUNTER

```

.*PRIORITY LEVEL DEFINITIONS

```

(1) 000000 PR0= 0 :: PRIORITY LEVEL 0
(1) 000040 PR1= 40 :: PRIORITY LEVEL 1
(1) 000100 PR2= 100 :: PRIORITY LEVEL 2
(1) 000140 PR3= 140 :: PRIORITY LEVEL 3
(1) 000200 PR4= 200 :: PRIORITY LEVEL 4
(1) 000240 PR5= 240 :: PRIORITY LEVEL 5
(1) 000300 PR6= 300 :: PRIORITY LEVEL 6
(1) 000340 PR7= 340 :: PRIORITY LEVEL 7

```

.*"SWITCH REGISTER" SWITCH DEFINITIONS

```

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

```

```

(1) .EQUIV SW01,SW1
(1) .EQUIV SW00,SW0
(1)
(1)
(1)
(1) 100000 .BIT15= 100000
(1) 040000 .BIT14= 40000
(1) 020000 .BIT13= 20000
(1) 010000 .BIT12= 10000
(1) 004000 .BIT11= 4000
(1) 002000 .BIT10= 2000
(1) 001000 .BIT09= 1000
(1) 000400 .BIT08= 400
(1) 000200 .BIT07= 200
(1) 000100 .BIT06= 100
(1) 000040 .BIT05= 40
(1) 000020 .BIT04= 20
(1) 000010 .BIT03= 10
(1) 000004 .BIT02= 4
(1) 000002 .BIT01= 2
(1) 000001 .BIT00= 1
(1) .EQUIV BIT09,BIT9
(1) .EQUIV BIT08,BIT8
(1) .EQUIV BIT07,BIT7
(1) .EQUIV BIT06,BIT6
(1) .EQUIV BIT05,BIT5
(1) .EQUIV BIT04,BIT4
(1) .EQUIV BIT03,BIT3
(1) .EQUIV BIT02,BIT2
(1) .EQUIV BIT01,BIT1
(1) .EQUIV BIT00,BIT0
(1)
(1) 000004 .: *BASIC "CPU" TRAP VECTOR ADDRESSES
(1) 000010 ERRVEC= 4 .: TIME OUT AND OTHER ERRORS
(1) 000014 RESVEC= 10 .: RESERVED AND ILLEGAL INSTRUCTIONS
(1) 000014 TBITVEC= 14 .: "T" BIT
(1) 000014 TRTVEC= 14 .: TRACE TRAP
(1) 000014 BPTVEC= 14 .: BREAKPOINT TRAP (BPT)
(1) 000020 IOTVEC= 20 .: INPUT/OUTPUT TRAP (IOT) **SCOPE**
(1) 000024 PWRVEC= 24 .: POWER FAIL
(1) 000030 EMTVEC= 30 .: EMULATOR TRAP (EMT) **ERROR**
(1) 000034 TRAPVEC= 34 .: "TRAP" TRAP
(1) 000060 TKVEC= 60 .: TTY KEYBOARD VECTOR
(1) 000064 TPVEC= 64 .: TTY PRINTER VECTOR
(1) 000240 PIRQVEC= 240 .: PROGRAM INTERRUPT REQUEST VECTOR
(1)
(1) ; *****
(1) .SBTTL RH11 REGISTERS
(1) ; *****
(1) ; CONTROL AND STATUS REGISTER 1 (RPCS1)
(1) 000100 IE= 100 .: INTERRUPT ENABLE (BIT #6)
(1) 000200 RDY= 200 .: READY (BIT #7)
(1) 000400 AH0= 400 .: HIGH ORDER BUS ADDRESS BIT (BIT #8)

```

```

7160      001000      A17=      1000      ;HIGH ORDER BUS ADDRESS BIT (BIT #9)
7161      002000      PSEL=      2000      ;PORT SELECT (BIT #10)
7162      020000      MCPE=      20000     ;MASSBUS PARITY ERROR (BIT #13)
7163      040000      TRE=       40000     ;TRANSFER ERROR (BIT #14)
7164      100000      SC=       100000     ;SPECIAL CONDITION (BIT #15)
7165
7166      ;WORD COUNT REGISTER (RPWC)
7167      ;(EACH BIT IS CALLED BY BIT NUMBER)
7168
7169      ;BUS ADDRESS REGISTER (RPBA)
7170      ;(EACH BIT IS CALLED BY BIT NUMBER)
7171
7172      ;CONTROL AND STATUS REGISTER 2 (RPCS2)
7173
7174      000001      US1=       1      ;UNIT SELECT (BIT #0)
7175      000002      US2=       2      ;UNIT SELECT (BIT #1)
7176      000004      US4=       4      ;UNIT SELECT (BIT #2)
7177      000010      BAI=      10      ;BUS ADDRESS INCREMENT INHIBIT (BIT #3)
7178      000020      PAT=      20      ;MASSBUS PARITY TEST (BIT #4)
7179      000040      CLR=      40      ;CLEAR (BIT #5)
7180      000100      IR=       100     ;INPUT READY (BIT #6)
7181      000200      OR=       200     ;OUTPUT READY (BIT #7)
7182      000400      MPE=      400     ;MASS BUS PARITY ERROR (BIT #8)
7183      001000      MXF=     1000     ;MISSED TRANSFER ERROR (BIT #9)
7184      002000      PGE=     2000     ;PROGRAM ERROR (BIT #10)
7185      004000      NEM=     4000     ;NON EXISTENT MEMORY (BIT #11)
7186      010000      NED=    10000     ;NON EXISTENT DRIVE (BIT #12)
7187      020000      UPE=    20000     ;UNIBUS PARITY ERROR (BIT #13)
7188      040000      WCE=    40000     ;WRITE CHECK ERROR (BIT #14)
7189      100000      DLT=   100000     ;DATA LATE (BIT #15)
7190
7191      ;DATA BUFFER REGISTER (RPDB)
7192      ;(EACH BIT IS CALLED BY BIT NUMBER)
7193
7194
7195      ;*****
7196
7197      .SBTTL  RP04/5/6 REGISTERS
7198
7199      ;*****
7200
7201      ;CONTROL AND STATUS 1 REGISTER. (#00)
7202
7203      000001      GO=       1      ;GO BIT (BIT #0)
7204      000002      F1=       2      ;FUNCTION CODE BIT #1
7205      000004      F2=       4      ;FUNCTION CODE BIT #2
7206      000010      F3=      10      ;FUNCTION CODE BIT #3
7207      000020      F4=      20      ;FUNCTION CODE BIT #4
7208      000040      F5=      40      ;FUNCTION CODE BIT #5
7209      004000      DVA=     4000     ;DEVICE AVAILABLE (BIT #11)
7210
7211      ;DRIVE STATUS REGISTER (RPDS1) (#01)
7212
7213      ;DFS=      1      DRIVE FORWARD 5"/SEC. (BIT #0)
7214      000002      DFF20=     2      ;DRIVE FORWARD 20"/SEC. (BIT #1)
7215      000004      DIGB=     4      ;DRIVE TO INNER GUARD BAND (BIT #2)

```


7216 000010
 7217 000020
 7218 000040
 7219 000100
 7220 000200
 7221 000400
 7222 001000
 7223 002000
 7224 004000
 7225 010000
 7226 020000
 7227 040000
 7228 100000

GRV= 10
 DL64= 20
 DE1= 40
 VV= 100
 DRY= 200
 DPR= 400
 PGM= 1000
 LST= 2000
 WRL= 4000
 MOL= 10000
 PIP= 20000
 ERR= 40000
 ATA= 100000

: 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)

7231 000001
 7232 000002
 7233 000004
 7234 000010
 7235 000020
 7236 000040
 7237 000100
 7238 000200
 7239 000400
 7240 001000
 7241 002000
 7242 004000
 7243 010000
 7244 020000
 7245 040000
 7246 100000

ILF= 1
 ILR= 2
 RMR= 4
 PAR= 10
 FER= 20
 WCF= 40
 ECH= 100
 HCE= 200
 HCRC= 400
 AOE= 1000
 IAE= 2000
 WLE= 4000
 DTE= 10000
 OPI= 20000
 UNS= 40000
 DCK= 100000

: 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)

7251 000001
 7252 000002
 7253 000004
 7254 000010
 7255 000020
 7256 000040
 7257 000200

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

: DIAGNOSTIC 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)

7261 000001
 7262 000002
 7263 000004
 7264 000010
 7265 000020
 7266 000040
 7267 000100
 7268 000200

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

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

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

7270
 7271

```

7272
7273 ;DRIVE TYPE REGISTER (RPDT) (#06)
7274
7275 000001 DT00= 1 ;DRIVE TYPE NUMBER BIT 1
7276 000002 DT01= 2 ;DRIVE TYPE NUMBER BIT 2
7277 000004 DT02= 4 ;DRIVE TYPE NUMBER BIT 3
7278 000010 DT03= 10 ;DRIVE TYPE NUMBER BIT 4
7279 000020 DT04= 20 ;DRIVE TYPE NUMBER BIT 5
7280 000040 DT05= 40 ;DRIVE TYPE NUMBER BIT 6
7281 000100 DT06= 100 ;DRIVE TYPE NUMBER BIT 7
7282 000200 DT07= 200 ;DRIVE TYPE NUMBER BIT 8
7283 000400 DT08= 400 ;DRIVE TYPE NUMBER BIT 9
7284 004000 DRQ= 4000 ;DRIVE REQUEST REQUIRED (BIT #11)
7285 020000 MOH= 20000 ;MOVING HEAD (BIT #13)
7286 040000 TAP= 40000 ;TAPE DRIVE (BIT #14)
7287 100000 NBA= 100000 ;NOT BLOCK ADDRESSED (BIT #15)
7288
7289 ;LOOK-AHEAD REGISTER (RPLA) (#07)
7290
7291 000001 EXT1= 1 ;EXTENSION 1 (BIT #0)
7292 000002 EXT2= 2 ;EXTENSION 2 (BIT #1)
7293 000004 EXT4= 4 ;EXTENSION 3 (BIT #2)
7294 000010 EXT10= 10 ;EXTENSION 4 (BIT #3)
7295 000020 EXT20= 20 ;EXTENSION 5 (BIT #4)
7296 000040 EXT40= 40 ;EXTENSION 6 (BIT #5)
7297 000100 SC1= 100 ;SECTOR COUNT FIELD 0 (BIT #6)
7298 000200 SC2= 200 ;SECTOR COUNT FIELD 1 (BIT #7)
7299 000400 SC4= 400 ;SECTOR COUNT FIELD 2 (BIT #8)
7300 001000 SC10= 1000 ;SECTOR COUNT FIELD 3 (BIT #9)
7301 002000 SC20= 2000 ;SECTOR COUNT FIELD 4 (BIT #10)
7302 004000 TRK1= 4000 ;TRACK FIELD 1 (BIT #11)
7303 010000 TRK2= 10000 ;TRACK FIELD 2 (BIT #12)
7304 020000 TRK4= 20000 ;TRACK FIELD 3 (BIT #13)
7305 040000 TRK10= 40000 ;TRACK FIELD 4 (BIT #14)
7306 100000 TRK20= 100000 ;TRACK FIELD 5 (BIT #15)
7307
7308 ;RPO4 ERROR REGISTER #2 (RPER2) (#10)
7309
7310 000001 WCU= 1 ;WRITE CURRENT UNSAFE (BIT #0)
7311 000002 CSF= 2 ;CURRENT SINK FAILURE (BIT #1)
7312 000004 WSU= 4 ;WRITE SELECT UNSAFE (BIT #2)
7313 000010 CSU= 10 ;CURRENT SWITCH UNSAFE (BIT #3)
7314 000020 MSE= 20 ;MOTOR SEQUENCE ERROR (BIT #4)
7315 000040 TOF= 40 ;TRANSITIONS DETECTOR FAILURE (BIT #5)
7316 000100 TJF= 100 ;TRANSITIONS UNSAFE (BIT #6)
7317 000200 FEN= 200 ;FAILSAFE ENABLED (BIT #7)
7318 000400 WRU= 400 ;WRITE READY UNSAFE (BIT #8)
7319 001000 MHS= 1000 ;MULTIPLE HEAD SELECT (BIT #9)
7320 002000 NHS= 2000 ;NO HEAD SELECTION (BIT #10)
7321 004000 IXE= 4000 ;INDEX ERROR (BIT #11)
7322 010000 VU30= 10000 ;30VOLT UNSAFE (BIT #12)
7323 020000 PLU= 20000 ;PLO UNSAFE (BIT #13)
7324 100000 ACU= 100000 ;AC UNSAFE (BIT #15)
7325
7326 ;RPO5/6 ERROR REGISTER #02 (RPER2) (#10)
7327
    
```

7328	000001	WCU=	1	:WRITE CURRENT UNSAFE (BIT #0)
7329	000002	CSF=	2	:CURRENT SINK FAILURE (BIT #1)
7330	000004	WSU=	4	:WRITE SELECT UNSAFE (BIT #2)
7331	000010	CSU=	10	:CURRENT SWITCH UNSAFE (BIT #3)
7332	000020	RAW=	20	:READ AND WRITE (BIT #4)
7333	000040	TDF=	40	:TRANSITIONS DETECTOR FAILURE (BIT #5)
7334	000100	TUF=	100	:TRANSITIONS UNSAFE (BIT #6)
7335	000200	ABS=	200	:ABNORMAL STOP (BIT #7)
7336	000400	WRU=	400	:WRITE READY UNSAFE (BIT #8)
7337	001000	MHS=	1000	:MULTIPLE HEAD SELECT (BIT #9)
7338	002000	NHS=	2000	:NO HEAD SELECTION (BIT #10)
7339	004000	IXE=	4000	:INDEX ERROR (BIT #11)
7340	020000	PLU=	20000	:PLO UNSAFE (BIT #12)
7341				
7342				
7343				
7344	000001	OF25=	1	:OFFSET 25 MICRO INCHES (BIT #0)
7345	000002	OF50=	2	:OFFSET 50 MICRO INCHES (BIT #1)
7346	000004	OF100=	4	:OFFSET 100 MICRO INCHES (BIT #2)
7347	000010	OF200=	10	:OFFSET 200 MICRO INCHES (BIT #3)
7348	000020	OF400=	20	:OFFSET 400 MICRO INCHES (BIT #4)
7349	000040	OF800=	40	:OFFSET 800 MICRO INCHES (BIT #5)
7350	000200	OFREV=	200	:OFFSET NEGATIVE (REVERSE) (BIT #5)
7351	002000	HCI=	2000	:HEADER COMPARE INHIBIT (BIT #10)
7352	004000	ECI=	4000	:ERROR CORRECTION CODE INHIBIT (BIT #11)
7353	010000	FMT22=	10000	:FORMAT BIT (BIT #12)
7354				
7355				
7356				
7357				
7358				
7359				
7360				
7361				
7362				
7363				
7364				
7365				
7366	000001	PSU=	1	:PACK SPEED UNSAFE (BIT #0)
7367	000002	VUF=	2	:VELOCITY UNSAFE (BIT #1)
7368	000010	UWR=	10	:ANY UNSAFE EXCEPT READ/WRITE (BIT #3)
7369	000020	PRE=	20	:DISK PACK ROTATION ERROR (BIT #4)
7370	000040	ACL=	40	:AC LOW (BIT #5)
7371	000100	DCL=	100	:DC LOW (BIT #6)
7372	040000	SKI=	40000	:SEEK INCOMPLETE (BIT #14)
7373	100000	OCYL=	100000	:OFF CYLINDER (BIT #15)
7374				
7375				
7376				
7377	000001	DCU=	1	:DC UNSAFE (BIT #0)
7378	000002	WAO=	2	:WRITE AND OFFSET (BIT #1)
7379	000040	ACL=	40	:AC LOW (BIT #5)
7380	000100	DCL=	100	:DC LOW (BIT #6)
7381	020000	OPE=	20000	:OPERATOR PLUG ERROR (BIT #13)
7382	040000	SKI=	40000	:SEEK INCOMPLETE (BIT #14)
7383	100000	OCYL=	100000	:OFF CYLINDER ERROR (BIT #15)

```

7384
7385
7386 ;ECC POSITION REGISTER (RPEC1) (#16)
7387 ;(EACH BIT IS CALLED BY BIT NUMBER)
7388
7389 ;ECC PATTERN REGISTER (RPEC2) (#17)
7390 ;(EACH BIT IS CALLED BY BIT NUMBER)
7391
7392 ;;*****
7393
7394 .SBTTL DEFINITIONS OF THE RH11/RP04/S/6 ADDRESS INDEXES
7395
7396 ;;*****
7397
7398 000000 RPCS1=0 ;CONTROL AND STATUS REGISTER #1 (DRIVE REG. 00)
7399 000002 RPWC=2 ;WORD COUNT REGISTER (NOT A DRIVE REG)
7400 000004 RPBA=4 ;UNIBUS ADDRESS REGISTER (NOT A DRIVE REG)
7401 000006 RPOA=6 ;DESIRED SECTOR/TRACK ADDRESS REGISTER (DRIVE REG. 05)
7402 000010 RPCS2=10 ;CONTROL AND STATUS REGISTER #2 (NOT A DRIVE REG)
7403 000012 RPS1=12 ;DRIVE STATUS REGISTER (DRIVE REG 01)
7404 000014 RPER1=14 ;ERROR REGISTER #1 (DRIVE REG. 02)
7405 000016 RPAS=16 ;ATTENTION SUMMARY PSEUDO REGISTER (DRIVE REG. 04)
7406 000020 RPLA=20 ;LOOK AHEAD REGISTER (DRIVE REG. 07)
7407 000022 RPOB=22 ;DATA BUFFER REGISTER (NOT A DRIVE REG.)
7408 000024 RPRM=24 ;MAINTAINABILITY REGISTER (DRIVE REG. 03)
7409 000026 RPDT=26 ;DRIVE TYPE REGISTER (DRIVE REG. 06)
7410 000030 RPSN=30 ;SERIAL NUMBER REGISTER (DRIVE REG. 10)
7411 000032 RPOF=32 ;OFFSET REGISTER (DRIVE REG. 11)
7412 000034 RPCA=34 ;DESIRED CYLINDER ADDRESS REGISTER (DRIVE REG. 12)
7413 000036 RPCC=36 ;CURRENT CYLINDER ADDRESS REGISTER (DRIVE REG. 13)
7414 000040 RPER2=40 ;ERROR REGISTER #2 (DRIVE REG. 14)
7415 000042 RPER3=42 ;ERROR REGISTER #3 (DRIVE REG. 15)
7416 000044 RPEC1=44 ;ECC POSITION REGISTER (DRIVE REG. 16)
7417 000046 RPEC2=46 ;ECC PATTERN REGISTER (DRIVE REG. 17)
7418
7419 .SBTTL TRAP CATCHER
(1)
(1) 000000 ;.=0
(1) ;*ALL UNUSED LOCATIONS FROM 4 - 776 CONTAIN A ".+2,HALT"
(1) ;*SEQUENCE TO CATCH ILLEGAL TRAPS AND INTERRUPTS
(1) ;*LOCATION 0 CONTAINS 0 TO CATCH IMPROPERLY LOADED VECTORS
(1) 000174 000174 ;.=174
(1) 000174 000000 DISPREG: .WORD 0 ;:SOFTWARE DISPLAY REGISTER
(1) 000176 000000 SWREG: .WORD 0 ;:SOFTWARE SWITCH REGISTER
7420
7421 .SBTTL ACT11 HOOKS
(1)
(2) ;;*****
(1) ;HOOKS REQUIRED BY ACT11
(1) 000200 $SVPC= ;SAVE PC
(1) 000046 ;.=46
(1) 000046 056454 $ENDAD ;:1)SET LOC.46 TO ADDRESS OF $ENDAD IN .SEOP
(1) 000052 ;.=52
(1) 000052 020000 .WORD 20000 ;:2)SET LOC.52 TO 20000
(1) 000200 ;.= $SVPC ;: RESTORE PC
7422

```

CZRJECO. DL CTRLR LGC MACY11 30A(1052) 28-DEC-77 10:17 PAGE 65-7
CZRJEC.P11 21-DEC-77 14:19 STARTING ADDRESS = 200

SEQ 0028

7423
7424
7425 000200 000137 002064
7426
7427
7428
7429 000204 000137 002072
7430
7472

.SBTTL STARTING ADDRESS = 200
JMP START ;START THE PROGRAM
.SBTTL START THE PROGRAM AND CHANGE THE RH11 ADDRESS = 204
JMP STAKT1 ;START AND CHANGE THE RH11 ADDRESS

7473

.SBTTL COMMON TAGS

```

(1)
(2) ;:*****
(1) ;:THIS TABLE CONTAINS VARIOUS COMMON STORAGE LOCATIONS
(1) ;:USED IN THE PROGRAM.
(1)
(1) 001100 001100          .SMTAG:      .=1100
(1) 001100 000000          SPASS:      .WORD      0          ;: START OF COMMON TAGS
(1) 001102 000          STSTNM:     .BYTE      0          ;: CONTAINS PASS COUNT
(1) 001103 000          SERFLG:     .BYTE      0          ;: CONTAINS THE TEST NUMBER
(1) 001104 000000          SICNT:      .WORD      0          ;: CONTAINS ERROR FLAG
(1) 001106 000000          SLPADR:     .WORD      0          ;: CONTAINS SUBTEST ITERATION COUNT
(1) 001110 000000          SLPERR:     .WORD      0          ;: CONTAINS SCOPE LOOP ADDRESS
(1) 001112 000000          SERTTL:     .WORD      0          ;: CONTAINS SCOPE RETURN FOR ERRORS
(1) 001114 000          SITEMB:     .BYTE      0          ;: CONTAINS TOTAL ERRORS DETECTED
(1) 001115 001          SERMAX:     .BYTE      1          ;: CONTAINS ITEM CONTROL E.TE
(1) 001116 000000          SERRPC:     .WORD      0          ;: CONTAINS MAX. ERRORS PER TEST
(1) 001120 000000          SGDADR:     .WORD      0          ;: CONTAINS PC OF LAST ERROR INSTRUCTION
(1) 001122 000000          SBDADR:     .WORD      0          ;: CONTAINS ADDRESS OF 'GOOD' DATA
(1) 001124 000000          SGGDAT:     .WORD      0          ;: CONTAINS ADDRESS OF 'BAD' DATA
(1) 001126 000000          SBODAT:     .WORD      0          ;: CONTAINS 'GOOD' DATA
(1) 001130 000000          .WORD      0          ;: CONTAINS 'BAD' DATA
(1) 001132 000000          .WORD      0          ;: RESERVED--NOT TO BE USED
(1) 001134 000          SAUTOB:     .BYTE      0          ;: AUTOMATIC MODE INDICATOR
(1) 001135 000          SINTAG:     .BYTE      0          ;: INTERRUPT MODE INDICATOR
(1) 001136 000000          .WORD      0          ;:
(1) 001140 177570          SWR:        .WORD      DSWR          ;: ADDRESS OF SWITCH REGISTER
(1) 001142 177570          DISPLAY:    .WORD      DDISP          ;: ADDRESS OF DISPLAY REGISTER
(1) 001144 177560          STKS:       177560          ;: TTY KBD STATUS
(1) 001146 177562          STKB:       177562          ;: TTY KBD BUFFER
(1) 001150 177564          STPS:       177564          ;: TTY PRINTER STATUS REG. ADDRESS
(1) 001152 177566          STPB:       177566          ;: TTY PRINTER BUFFER REG. ADDRESS
(1) 001154 000          SNULL:      .BYTE      0          ;: CONTAINS NULL CHARACTER FOR FILLS
(1) 001155 002          SFILLS:     .BYTE      2          ;: CONTAINS # OF FILLER CHARACTERS REQUIRED
(1) 001156 012          SFILLC:     .BYTE      12         ;: INSERT FILL CHARS. AFTER A "LINE FEED"
(1) 001157 000          STPFLG:     .BYTE      0          ;: "TERMINAL AVAILABLE" FLAG (BIT<07>=0=YES
(1) 001160 000000          SREGAD:     .WORD      0          ;: CONTAINS THE ADDRESS FROM WHICH ($REGD) WAS OBTAINED
(3) 001162 000000          SREGO:      .WORD      0          ;: CONTAINS (($REGAD)+0)
(3) 001164 000000          STMP0:      .WORD      0          ;: USER DEFINED
(3) 001166 000000          STMP1:      .WORD      0          ;: USER DEFINED
(3) 001170 000000          STMP2:      .WORD      0          ;: USER DEFINED
(3) 001172 000000          STMP3:      .WORD      0          ;: USER DEFINED
(3) 001174 000000          STMP4:      .WORD      0          ;: USER DEFINED
(1) 001176 000000          STIMES:     0          ;: MAX. NUMBER OF ITERATIONS
(1) 001200 000000          SESCAPE:    0          ;: ESCAPE ON ERROR ADDRESS
(1) 001202 177607 000377          SBELL:      .ASCIZ    <207><377><377> ;: CODE FOR BELL
(1) 001206 077          SQUES:      .ASCII    /?/          ;: QUESTION MARK
(1) 001207 015          SCRLF:      .ASCII    <15>          ;: CARRIAGE RETURN
(1) 001210 000012          SLF:        .ASCIZ    <12>          ;: LINE FEED
(2) ;:*****
(3) CR = 15
(3) LF = 12
(3) 001212 172540          $LKCSR:     .WORD      172540        ;: ADDR OF KW11-P STATUS REGISTER
(3) 001214 172542          $LKCSB:     .WORD      172542        ;: ADDR OF KW11-P COUNTER BUFFER

```

(3)	001216	000104	\$LPVEC: .WORD	104	; ADDR OF KW11-P VECTOR
(3)	001220	177546	\$LKS: .WORD	177546	; ADDR OF KW11-L STATUS REGISTER
(3)	001222	000100	\$LLVEC: .WORD	100	; ADDR OF KW11-L VECTOR
(3)	001224	000000	PORTA: .WORD	0	; ADDRESS OF PORT A
(3)	001226	000000	PORTB: .WORD	0	; ADDRESS OF PORT B
(3)	001230	000000	PORTC: .WORD	0	; ADDRESS OF DIFFERENT DRIVE
(3)	001232	000000	ASR1: .WORD	0	; ATA-A OR ATA-B = 1
(3)	001234	000000	PTNBR: .WORD	0	; CONTAINS THE PORT ADDRESS FOR ERROR TYPEOUTS
(3)	001236	000000	SEIZPT: .WORD	0	; CONTAINS THE ADDRESS OF THE SEIZING PORT
(3)	001240	000000	OPPR: .WORD	0	; CONTAINS THE ADDRESS OF THE 'OPPOSITE' PORT
(3)	001242	000000	TSTNUM: .WORD	0	; NUMBER OF THE CURRENT TEST
(3)	001244	000000	CKERR: .WORD	0	; IF -1, A REGISTER MISCOMPARISON OCCURRED
(3)	001246	000000	NOSEIZ: .WORD	0	; IF -1, THE PORT IN 'SEIZPT' DID NOT SEIZE THE DRIVE
(3)	001250	000000	RELER: .WORD	0	; IF -1, THE PORT IN 'SEIZPT' DID NOT RELEASE THE DRIVE
(3)	001252	000000	TIME: .WORD	0	; ELAPSED TIME COUNTER
(3)	001254	000000	WATCH: .WORD	0	; WATCH DOG TIMER LOCATION
(3)	001256	000000	TIMEA: .WORD	0	; THE TIMEOUT ONE-SHOT VALUE MEASURED THROUGH PORT A
(3)	001260	000000	TIMEAP: .WORD	0	; PORT A TIMEOUT VALUE + 25%
(3)	001262	000000	TIMEAM: .WORD	0	; PORT A TIMEOUT VALUE - 25%
(3)	001264	000000	TIMEB: .WORD	0	; THE TIMEOUT ONE-SHOT VALUE MEASURED THROUGH PORT B
(3)	001266	000000	TIMEBP: .WORD	0	; PORT B TIMEOUT VALUE + 25%
(3)	001270	000000	TIMEBM: .WORD	0	; PORT B TIMEOUT VALUE - 25%
(3)	001272	000000	TIMES: .WORD	0	; STORAGE FOR TIMEOUT ONE-SHOT RETRIGGER TEST
(3)	001274	000000	KYBCTL: .WORD	0	; SINGLE TEST INDICATOR
(3)	001276	000000	CHGADR: .WORD	0	; CHANGE THE RH11 ADDRESS INDICATOR

::*****

.SBTTL RH11/RP04/5/6 UNIBUS AND VECTOR ADDRESSES

::*****

(3)	001300	176700	\$RPADR: .WORD	176700	; RH11/RP04/5/6 UNIBUS ADDRESS
(3)	001302	000254	\$RPVEC: .WORD	254	; RH11 INTERRUPT VECTOR ADDRESS

7515	001362	070577	DFS	
7516				
7517				:ERROR 7
7518				
7519	001364	063052	EM7	:REGISTER CONTENTS INCORRECT AFTER RELEASE/TIMEOUT
7520	001366	067103	DH7	
7521	001370	070404	DT7	
7522	001372	070605	DF7	
7523				
7524				:ERROR 10
7525				
7526	001374	063133	EM10	:REGISTER CONTENTS INCORRECT
7527	001376	067027	DH5	
7528	001400	070352	DT5	
7529	001402	070577	DFS	
7530				
7531				:ERROR 11
7532				
7533	001404	063163	EM11	:CONTROL BUS PARITY ERROR WHILE READING REGISTER
7534	001406	067226	DH11	
7535	001410	070322	DT1	
7536	001412	070572	DF1	
7537				
7538				:ERROR 12
7539				
7540	001414	063247	EM12	:DRIVE NOT SEIZED BY DRIVE CLEAR COMMAND
7541	001416	067773	DH36	
7542	001420	070512	DT37	
7543	001422	070620	DF36	
7544				
7545				:ERROR 13
7546				
7547	001424	063317	EM13	: 'VOLUME VALID' BIT NOT SET BY READIN PRESET
7548	001426	067277	DH13	
7549	001430	070424	DT13	
7550	001432	070577	DFS	
7551				
7552				:ERROR 14
7553				
7554	001434	063404	EM14	: 'VOLUME VALID' SET ON THE OPPOSITE PORT
7555	001436	067277	DH13	
7556	001440	070424	DT13	
7557	001442	070577	DFS	
7558				
7559				:ERROR 15
7560				
7561	001444	063447	EM15	:THE ATTN BIT WRONG AFTER TIMEOUT - REQUEST NOT SET
7562	001446	067103	DH7	
7563	001450	070404	DT7	
7564	001452	070605	DF7	
7565				
7566				:ERROR 16
7567				
7568	001454	063526	EM16	:ATTN BIT WRONG AFTER RELEASE - REQUEST WAS SET
7569	001456	067103	DH7	
7570	001460	070404	DT7	

7571	001462	070605	DF7	
7572				
7573				;ERROR 17
7574				
7575	001464	063601	EM17	;ATTN BIT WRONG AFTER RELEASE - REQUEST NOT SET
7576	001466	067103	DH7	
7577	001470	070404	DT7	
7578	001472	070605	DF7	
7579				
7580				;ERROR 20
7581				
7582	001474	063660	EM20	;DRIVE NOT SEIZED WHEN ATTN BIT FOR PORT CLEARED
7583	001476	067773	DH36	
7584	001500	070512	DT37	
7585	001502	070620	DF36	
7586				
7587				;ERROR 21
7588				
7589	001504	063740	EM21	;DRIVE SEIZED WHEN ZERO WRITTEN IN ATTN BIT FOR PORT
7590	001506	067773	DH36	
7591	001510	070512	DT37	
7592	001512	070620	DF36	
7593				
7594				;ERROR 22
7595				
7596	001514	064013	EM22	;DRIVE NOT IN NEUTRAL AFTER TIMEOUT, REQUEST NOT SET
7597	001516	067417	DH22	
7598	001520	070442	DT22	
7599	001522	070614	DF31	
7600				
7601				;ERROR 23
7602				
7603	001524	064100	EM23	;TIMEOUT CLEARED THE DRIVE'S ERROR BIT
7604	001526	067515	DH23	
7605	001530	070454	DT23	
7606	001532	070572	DF1	
7607				

7609			:ERROR 24	
7610				
7611	001534	064146	EM24	:RELEASE COMMAND RELEASED DRIVE WITH ERRORS SET
7612	001536	067515	DH23	
7613	001540	070454	DT23	
7614	001542	070572	DF1	
7615				
7616				
7617			:ERROR 25	
7618				
7619	001544	064225	EM25	:TIMEOUT ONE-SHOT DID NOT PETRIGGER
7620	001546	067773	DH36	
7621	001550	070502	DT36	
7622	001552	070620	DF36	
7623				
7624				
7625			:ERROR 26	
7626				
7627	001554	064270	EM26	:DRIVE NOT IN NEUTRAL AFTER RELEASE. REQUEST NOT SET
7628	001556	067417	DH22	
7629	001560	070442	DT22	
7630	001562	070614	DF31	
7631				
7632			:ERROR 27	
7633				
7634	001564	064355	EM27	:REGISTER WRONG AFTER RELEASE WITH REQUEST SET
7635	001566	067103	DH7	
7636	001570	070404	DT7	
7637	001572	070605	DF7	
7638				
7639			:ERROR 30	
7640				
7641	001574	064433	EM30	:DRIVE SEIZED BY RELEASE ISSUED WHEN DRIVE IN NEUTRAL
7642	001576	067773	DH36	
7643	001600	070502	DT36	
7644	001602	070620	DF36	
7645				
7646			:ERROR 31	
7647				
7648	001604	064530	EM31	:DRIVE NOT SEIZED BY PORT AFTER RELEASE WITH REQUEST SE
7649	001606	067674	DH31	
7650	001610	070470	DT31	
7651	001612	070614	DF31	
7652				
7653			:ERROR 32	
7654				
7655	001614	064605	EM32	:ATTN BIT WRONG AFTER RECALIBRATE COMMAND
7656	001616	067027	DH5	
7657	001620	070352	DT5	
7658	001622	070577	DF5	
7659				
7660			:ERROR 33	
7661				
7662	001624	064656	EM33	:DRIVE RETURNS TO NEUTRAL IF DRIVE CLEAR GIVEN WHILE DRI
7663	001626	067773	DH36	
7664	001630	070502	DT36	

7665	001632	070620	DF36	
7666				
7667				; ERROR 34
7668				
7669	001634	064760	EM34	; DRIVE RETURNS TO NEUTRAL IF MASSBUS INIT GIVEN WHILE DR
7670	001636	067773	DH36	
7671	001640	070502	DT36	
7672	001642	070620	DF36	
7673				
7674				; ERROR 35
7675				
7676	001644	065063	EM35	; DRIVE RETURNED TO NEUTRAL WITHOUT TRIGGERING TIMEOUT ON
7677	001646	067773	DH36	
7678	001650	070512	DT37	
7679	001652	070620	DF36	
7680				
7681				; ERROR 36
7682				
7683	001654	065142	EM36	; TIMEOUT HAS NOT OCCURRED WITHIN 2 SECONDS
7684	001656	067773	DH36	
7685	001660	070502	DT36	
7686	001662	070620	DF36	
7687				
7688				; ERROR 37
7689				
7690	001664	065214	EM37	; DRIVE IS NON-EXISTENT
7691	001666	067773	DH36	
7692	001670	070512	DT37	
7693	001672	070620	DF36	
7694				
7695				; ERROR 40
7696				
7697	001674	065262	EM40	; ATTENTION FOR PORT NOT RESET BY MASSBUS CLEAR
7698	001676	066564	DH1	
7699	001700	070454	DT23	
7700	001702	070572	DF1	
7701				
7702				; ERROR 41
7703				
7704	001704	065337	EM41	; TIMEOUT CLEARED ATTENTION BIT
7705	001706	067515	DH23	
7706	001710	070454	DT23	
7707	001712	070572	DF1	
7708				
7709				; ERROR 42
7710				
7711	001714	065401	EM42	; DRIVE NOT IN NEUTRAL OR SEIZED
7712	001716	070022	DH42	
7713	001720	070522	DT42	
7714	001722	070623	DF42	
7715				
7716				; ERROR 43
7717				
7718	001724	065467	EM43	; DRIVE IN NEUTRAL AFTER ATTENTION BIT WRITTEN
7719	001726	070022	DH42	
7720	001730	070522	DT42	

7721	001732	070623	DF42	
7722				
7723				;ERROR 44
7724				
7725	001734	065544	EM44	;WRITE ATTENTION BIT DID NOT SET PORT REQUEST
7726	001736	070041	DH44	
7727	001740	070470	DT31	
7728	001742	070614	DF31	
7729				
7730				;ERROR 45
7731				
7732	001744	065621	EM45	;CONTROLLER SELECT SWITCH ON DRIVE NOT IN 'A/B'
7733	001746	066564	DH1	
7734	001750	070322	DT1	
7735	001752	070572	DF1	
7736				
7737				;ERROR 46
7738				
7739	001754	065700	EM46	;CAN'T ACCESS DRIVE THROUGH EITHER PORT
7740	001756	070137	DH46	
7741	001760	070530	DT46	
7742	001762	070614	DF31	
7743				
7744				;ERROR 47
7745				
7746	001764	065747	EM47	;ATTN BIT FOR SEIZING PORT NOT CLEARED BY MASSBUS INIT
7747	001766	067515	DH23	
7748	001770	070454	DT23	
7749	001772	070572	DF1	
7750				
7751				;ERROR 50
7752				
7753	001774	066035	EM50	;ATTN BIT FOR OPPOSITE PORT CLEARED BY MASSBUS INIT
7754	001776	067277	DH13	
7755	002000	070424	DT13	
7756	002002	070577	DF5	
7757				
7758				;ERROR 51
7759				
7760	002004	066120	EM51	;ATTN BIT CLEARED BY MASSBUS INIT. DRIVE IN NEUTRAL
7761	002006	067027	DH5	
7762	002010	070352	DT5	
7763	002012	070577	DF5	
7764				
7765				;ERROR 52
7766				
7767	002014	066203	EM52	;ATTN BIT SET AFTER TIMEOUT, 'ERR' SET. NO REQUEST
7768	002016	067277	DH13	
7769	002020	070424	DT13	
7770	002022	070577	DF5	
7771				
7772				;ERROR 53
7773				
7774	002024	066301	EM53	
7775	002026	067515	DH23	
7776	002030	070322	DT1	

```

7777 002032 070572          DF1
7778
7779          ;ERROR 54
7780
7781 002034 066362          EM54          ;RELEASE COMMAND RECOGNIZED WHEN ISSUED BY NON-SEIZING P
7782 002036 067417          DH22
7783 002040 070542          DT54
7784 002042 070614          DF31
7785
7786          ;ERROR 55
7787
7788 002044 066455          EM55          ;TIMEOUT ONE-SHOT IS LESS THAN 500 MS
7789 002046 070235          DH55
7790 002050 070554          DT55
7791 002052 070625          DF55
7792
7793          ;ERROR 56
7794
7795 002054 066522          EM56          ;RH11 DIDN'T RESPOND TO ADDRESSING
7796 002056 070313          DH56
7797 002060 070566          DT56
7798 002062 070631          DF56
7799
7800
7801
7808          ;*****
7809          .SBTTL  STARTUP AND INITIALIZATION ROUTINES
7810
7811          ;*****
7812
7813
7814 002064 005037 001276    START:  CLR      CHGADR          ;CLEAR THE 'CHANGE RH11 ADDRESS' INDICATOR
7815 002070 000403          BR      START2          ;GO TO THE START
7816 002072 012737 177777 001276  START1: MOV      #-1,CHGADR      ;SET THE 'CHANGE RH11 ADDRESS' INDICATOR
7817 002100 000005          START2: RESET          ;CLEAR THE BUS
7818          .SBTTL  INITIALIZE THE COMMON TAGS
(1)          ;:CLEAR THE COMMON TAGS (%CMTAG) AREA
(1) 002102 012706 001100    MOV      %SCMTAG,R6      ;:FIRST LOCATION TO BE CLEARED
(1) 002106 005026          CLR      (R6)+          ;:CLEAR MEMORY LOCATION
(1) 002110 022706 001140    CMP      %SWR,R6 ;:DONE?
(1) 002114 001374          BNE     -.6             ;:LOOP BACK IF NO
(1) 002116 012706 001100    MOV      %STACK,SP      ;:SETUP THE STACK POINTER
(1)          ;:INITIALIZE A FEW VECTORS
(1) 002122 012737 056720 000020  MOV      %$SCOPE,%IOTVEC ;:IOT VECTOR FOR SCOPE ROUTINE
(1) 002130 012737 000340 000022  MOV      %340,%IOTVEC+2 ;:LEVEL 7
(1) 002136 012737 057152 000030  MOV      %$ERROR,%EMTVEC ;:EMT VECTOR FOR ERROR ROUTINE
(1) 002144 012737 000340 000032  MOV      %340,%EMTVEC+2 ;:LEVEL 7
(1) 002152 012737 061752 000034  MOV      %$TRAP,%TRAPVEC ;:TRAP VECTOR FOR TRAP CALLS
(1) 002160 012737 000340 000036  MOV      %340,%TRAPVEC+2 ;:LEVEL 7
(1) 002166 013737 056320 056312  MOV      %ENDCT,%EOPCT  ;:SETUP END-OF-PROGRAM COUNTER
(1) 002174 005037 001176          CLR      %TIMES         ;:INITIALIZE NUMBER OF ITERATIONS
(1) 002200 005037 001200          CLR      %ESCAPE        ;:CLEAR THE ESCAPE ON ERROR ADDRESS
(1) 002204 112737 000001 001115  MOVB    #1,%SERMAX      ;:ALLOW ONE ERROR PER TEST
(1) 002212 012737 002212 001106  MOV     #,%$LPADR       ;:INITIALIZE THE LOOP ADDRESS FOR SCOPE
(1) 002220 012737 002220 001110  MOV     #,%$LPERR       ;:SETUP THE ERROR LOOP ADDRESS
(2)          ;:SIZE FOR A HARDWARE SWITCH REGISTER. IF NOT FOUND OR IT IS

```

```

(2)          ;;EQUAL TO A "-1" SETUP FOR A SOFTWARE SWITCH REGISTER.
(2) 002226 013746 000004          MOV    2#ERRVEC, -(SP)  ;;SAVE ERROR VECTOR
(2) 002232 012737 002266 000004  MOV    #64$, 2#ERRVEC  ;;SET UP ERROR VECTOR
(2) 002240 012737 177570 001140  MOV    #DSWR, SWR      ;;SETUP FOR A HARDWARE SWICH REGISTER
(2) 002246 012737 177570 001142  MOV    #DDISP, DISPLAY ;;AND A HARDWARE DISPLAY REGISTER
(2) 002254 022777 177777 176656  CMP    #-1, 2#SWR     ;;TRY TO REFERENCE HARDWARE SWR
(2) 002262 001012          BNE    66$            ;;BRANCH IF NO TIMEOUT TRAP OCCURRED
(2)          ;;AND THE HARDWARE SWR IS NOT = -1
(2) 002264 000403          BR     65$            ;;BRANCH IF NO TIMEOUT
(2) 002266 012716 002274          MOV    #65$, (SP)    ;;SET UP FOR TRAP RETURN
(2) 002272 000002          RTI
(2) 002274 012737 000176 001140 65$:  MOV    #SWREG, SWR    ;;POINT TO SOFTWARE SWR
(2) 002302 012737 000174 001142  MOV    #DISPREG, DISPLAY
(2) 002310 012637 000004 66$:  MOV    (SP)+, 2#ERRVEC ;;RESTORE ERROR VECTOR
(1)
7819 002314 005227 177777          INC    #-1           ;;FIRST START ?
7820 002320 001006          BNE    1$           ;;BR IF NOT
7821 002322 023737 000042 000046  CMP    2#42, 2#46   ;;ACT11?
7822 002330 001402          BEQ    1$           ;;BR IF YES
7823 002332 104401 062040          TYPE  TITLE        ;;TYPE PROGRAM NAME
7824 002336 004737 060362          JSR    PC, $TKINT   ;;SETUP THE TTY KEYBOARD
7825          .SBTTL GET VALUE FOR SOFTWARE SWITCH REGISTER
(1) 002342 005737 000042          TST    2#42         ;;ARE WE RUNNING UNDER XXDP/ACT?
(1) 002346 001006          BNE    67$         ;;BRANCH IF YES
(1) 002350 023727 001140 000176  CMP    SWR, #SWREG  ;;SOFTWARE SWITCH REG SELECTED?
(1) 002356 001005          BNE    68$         ;;BRANCH IF NO
(1) 002360 104406          GTSWR              ;;GET SOFT-SWR SETTINGS
(1) 002362 000403          BR     68$
(1) 002364 112737 000001 001134 67$:  MOVB   #1, $AUTOB  ;;SET AUTO-MODE INDICATOR
(1) 002372          68$:
7826 002372 004737 003004          JSR    PC, CHANGE   ;;CHECK/CHANGE THE RH11 ADDRESS
7827 002376 023737 000042 000046  CMP    2#42, 2#46   ;;ACT11?
7828 002404 001414          BEQ    2$           ;;BR IF YES
7829 002406 104401 062137          TYPE  ,ENTERA      ;;ENTER DRIVE ADDRESS
7830 002412 104412          RDOCT              ;;GET THE ADDRESS
7831 002414 012637 001224          MOV    (SP)+, PORTA ;;STORE THE ADDRESS
7832 002420 023727 001224 000007  CMP    PORTA, #7    ;;SEE IF ADDRESS TOO LARGE
7833 002426 101403          BLOS   2$           ;;BR IF NOT
7834 002430 104401 062167          TYPE  ADRERR       ;;TYPE ADDRESS ERROR MESSAGE
7835 002434 000740          BR     1$           ;;TRY AGAIN
7836 002436 013737 001224 001226 2$:  MOV    PORTA, PORTB ;;GENERATE THE PORT B ADDRESS
7837 002444 005237 001226          INC    PORTB        ;;INCREMENT THE ADDRESS
7838 002450 042737 000016 001226  BIC    #16, PORTB   ;;LEAVE BIT 0
7839 002456 013746 001224          MOV    PORTA, -(SP) ;;PUT PORT A ADDRESS ON THE STACK
7840 002462 042716 177771          BIC    #1C6, (SP)  ;;SAVE BITS 1 & 2
7841 002466 052637 001226          BIS    (SP)+, PORTB ;;SET BITS 1 & 2 IN PORT B ADDRESS
7842 002472 104401 062211          TYPE  PORTAIS      ;;'PORT A ADDRESS IS '
7843 002476 013746 001224          MOV    PORTA, -(SP) ;;SAVE PORTA FOR TYPEOUT
(1)          ;;TYPE PORT A ADDRESS
(1) 002502 104403          TYPOS              ;;GO TYPE--OCTAL ASCII
(1) 002504 001          .BYTE 1           ;;TYPE 1 DIGIT(S)
(1) 002505 000          .BYTE 0           ;;SUPPRESS LEADING ZEROS
7844 002506 104401 062237          TYPE  PORTBIS      ;;'PORT B ADDRESS IS '
7845 002512 013746 001226          MOV    PORTB, -(SP) ;;SAVE PORTB FOR TYPEOUT
(1)          ;;TYPE PORT B ADDRESS
(1) 002516 104403          TYPOS              ;;GO TYPE--OCTAL ASCII

```

N03

```

(1) 002520 001 .BYTE 1 ;:TYPE 1 DIGIT(S)
(1) 002521 000 .BYTE 0 ;:SUPPRESS LEADING ZEROS
7846 002522 104401 001207 TYPE $CRLF ;:ANOTHER CR-LF
7847 002526 013737 001224 001230 MOV PORTA,PORTC ;:GENERATE ADDRESS OF DRIVE NOT TESTED
7848 002534 062737 000006 001230 ADD #6,PORTC ;:COMPLEMENT SOME BITS
7849 002542 042737 177770 001230 BIC #+C7,PORTC ;:SAVE ONLY LOWER BITS
7850 002550 013701 001224 MOV PORTA,R1 ;:USE PORT A ADDRESS AS INDEX
7851 002554 116137 070746 001232 MOVB ATABIT(R1),ASR1 ;:GET ATTENTION BIT FOR DRIVE
7854 002562 005037 001256 CLR TIMEA ;:CLEAR TIMEOUT ONE-SHOT VALUE LOCATION
(1) 002566 005037 001260 CLR TIMEAP ;:CLEAR TIMEOUT ONE-SHOT VALUE LOCATION
(1) 002572 005037 001264 CLR TIMEB ;:CLEAR TIMEOUT ONE-SHOT VALUE LOCATION
(1) 002576 005037 001266 CLR TIMEBP ;:CLEAR TIMEOUT ONE-SHOT VALUE LOCATION
7855 002602 004737 056474 JSR PC,CKCLK ;:SETUP CLOCK
7856 002606 000137 002622 JMP EXEC ;:CLOCK HAS BEEN STARTED
7857 002612 104401 062265 TYPE ,NOCLOCK ;:NO CLOCK ON SYSTEM
7858 002616 000000 3$: HALT ;:FATAL ERROR
7859 002620 000776 BR 3$ ;:INTERLOCK THE HALT
7860
7861 ;ROUTINE TO GET THE TEST NUMBER FROM THE OPERATOR
7862
7863 002622 000005 EXEC: RESET ;:CLEAR EVERYTHING
7864 002624 005037 177776 CLR PS ;:CLEAR THE PROCESSOR STATUS WORD
7865 002630 104401 001207 TYPE $CRLF ;:CR-LF
7866 002634 013700 001300 MOV $RPADR,RO ;:RH11 ADDRESS FOR INDEXING
7867 002640 012706 001100 MOV #STACK,SP ;:LOAD STACK POINTER
7868 002644 004737 056474 JSR PC,CKCLK ;:START THE CLOCK
7869 002650 000240 NOP ;:RETURN IF NO CLOCK
7870 002652 004737 060362 JSR PC,STKINT ;:INITIALIZE THE KEYBOARD
7871 002656 005037 001274 CLR KYBCTL ;:CLEAR SINGLE TEST INDICATOR
7872 002662 005037 001100 CLR $PASS ;:CLEAR THE PASS COUNT
7873 002666 112737 000001 001115 MOVB #1,$ERMAX ;:SET ERROR MAX TO 1
7874 002674 012737 002674 001106 MOV #,$SLPADR ;:INITIAL SETTING FOR LOOP ADDRESS
7875 002702 012737 002702 001110 MOV #,$SLPERR ;:INITIAL SETTING FOR LOOP ON ERROR ADDRESS
7876 002710 023737 000042 000046 1$: CMP #42,#46 ;:ACT11?
7877 002716 001405 BEQ 5$ ;:BR IF YES
7878 002720 104401 062334 TYPE ,TESTNO ;:ASK FOR TEST NUMBER
7879 002724 104412 RDOCT ;:GET THE NUMBER
7880 002726 012601 MOV (SP)+,R1 ;:PUT ENTRY INTO R1
7881 002730 001002 BNE 2$ ;:BR IF NOT ZERO
7882 002732 000137 003120 5$: JMP TST1 ;:ENTER ZERO - PERFORM ALL TESTS
7883 002736 020137 070756 2$: CMP R1,MAXTN ;:SEE IF NUMBER GREATER THAN MAXIMUM
7884 002742 003403 BLE 3$ ;:BR IF LESS OR EQUAL
7885 002744 104401 062354 TYPE ,BADNO ;:BAD ENTRY
7886 002750 000757 BR 1$ ;:TRY AGAIN
7887 002752 005301 3$: DEC R1 ;:DECREMENT ENTRY
7888 002754 006301 ASL R1 ;:SHIFT IT LEFT
7889 002756 016137 070632 003002 MOV TSTADR(R1),4$ ;:GET THE TEST ADDRESS
7890 002764 005237 001274 INC KYBCTL ;:SET SINGLE TEST INDICATOR
7891 002770 012737 000001 001104 MOV #1,$ICNT ;:PRESET ITERATION COUNT
7892 002776 000177 000000 JMP #4$ ;:GO TO THE SELECTED TEST
7893 003002 000000 4$: .WORD 0 ;:TEST ADDRESS GOES HERE
7894
7895 ;CHANGE THE RH11 UNIBUS ADDRESS USED BY THE PROGRAM
7896
7897 003004 005737 001276 CHANGE: TST CHGADR ;:CHANGE THE ADDRESS ?
7898 003010 001421 BEQ 3$ ;:BR IF NOT
    
```



```

7899 003012 005037 001276          CLR      CHGADR      ;CLEAR THE INDICATOR
7900 003016 104401 062414          1$:     TYPE      ADDRIS ;TYPE OUT WHAT THE PRESENT ADDRESS IS
7901 003022 013746 001300          MOV      $RPADR,-(SP) ;PUT THE ADDRESS ON THE STACK
7902 003026 104402          TYP0C      ;TYPE THE ACTUAL ADDRESS
7903 003030 104401 001207          TYPE      ,$CRLF      ;CR-LF
7904 003034 104401 062474          TYPE      ,NTRH11     ;ASK FOR NEW ADDRESS
7905 003040 104412          RDOCT      ;
7906 003042 005716          TST      (SP)         ;O OR 'CR' ENTERED ?
7907 003044 001402          BEQ      2$          ;BR IF EITHER ENTERED (NO ADDRESS CHANGE)
7908 003046 011637 001300          MOV      (SP), $RPADR ;NEW RH11 ADDRESS
7909 003052 005726          2$:     TST      (SP)+ ;CORRECT THE STACK POINTER
7910 003054 012737 003074 000004 3$:     MOV      #4,$J#4   ;LOAD TRAP ADDRESS
7911 003062 013700 001300          MOV      $RPADR,RO   ;RH11 ADDRESS
7912 003066 005760 000002          TST      RPWC(RO)    ;SEE IF RH11 RESPONDS AT THAT ADDRESS
7913 003072 000404          BR       5$          ;BR, RH11 ALIVE AT PRESENT ADDRESS
7914 003074 104056          4$:     ERROR     56   ;NO RESPONSE TO ADDRESS
7915 003076 062706 000004          ADD      #4,SP       ;RESET THE STACK POINTER
7916 003102 000745          BR       1$          ;GET ADDRESS AGAIN
7917 003104 012737 000006 000004 5$:     MOV      #6,$J#4   ;RESTORE THE VECTOR
7918 003112 000207          RTS      PC          ;RETURN

```

::*****

.SBTTL *** TESTS ***

::*****

```

7938
7939 003114 013700 001300          TST1AA: MOV      $RPADR,RO ;;RESTORE RO AFTER END OF PASS
7940

```

::*****

```

(3) *TEST 1      DRIVE ACCESS TEST
(4) *
(4) *VERIFY THAT THE DRIVE CAN BE ACCESSED THROUGH BOTH PORTS
(4) *
(4) *   A.  SELECT DRIVE, VERIFY THAT THE DRIVE IS PRESENT, THAT THE
(4) *       DRIVE IS A DUAL PORT RPO4, THAT THE DRIVE IS ONLINE (RPDS1 HAS
(4) *       'MOL' 'PGM' 'DPR', & 'DRY' BITS SET), AND THE THE DRIVE SERIAL
(4) *       NUMBER READ THROUGH BOTH PORTS IS THE SAME.
(4) *
(4) *   B.  THE TEST IS REPEATED THROUGH BOTH PORTS.
(3) *

```

::*****

```

(2)
(3) 003120          TST1:
(3) 003120 005737 001274          TST      KYBCTL      ;PERFORMING ONLY SINGLE TESTS ?
(3) 003124 001406          BEQ      2$          ;BR IF NOT
(3) 003126 100002          BPL      1$          ;BR IF JUST ENTERED TEST
(3) 003130 000137 002622          JMP      EXEC        ;RETURN & GET NEXT TEST NUMBER
(3) 003134 012737 177777 001274 1$:     MOV      #-1,KYBCTL ;SET SINGLE TEST INDICATOR
(3) 003142 112737 000001 001102 2$:     MOV      #1,$STNM   ;TEST NUMBER
(3) 003150 012737 003172 001106          MOV      #TEST1,$LPADR ;LOAD LOOP ON TEST ADDRESS
(3) 003156 012737 003172 001110          MOV      #TEST1,$LPERR ;LOAD LOOP ON ERROR ADDRESS
(1) 003164 012737 000001 001176          MOV      #1,$TIMES   ;DO 1 ITERATION
7942 002172 012706 001100          TEST1: MOV      #STACK,SP ;LOAD THE STACK POINTER

```

::*****

;VERIFY THAT DRIVE IS PRESENT THROUGH PORTS A & B

```

7945
7946
7954 003176 113760 001224 000010      MOV      PORTA,RPCS2(RO) ;SELECT PORT A
      (2) 003204 013737 001224 001234      MOV      PORTA,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
      (1) 003212 005760 000012              TST      RPS1(RO) ;SEE IF DRIVE (PORT A) PRESENT
      (2) 003216 005037 001244              CLR      CKERR ;CLEAR THE 'CHECK ERROR' INDICATOR
      (2) 003222 016037 000010 001126      MOV      RPCS2(RO),%BDDAT ;GET CONTENTS OF RPCS2
      (2) 003230 012737 000010 001122      MOV      %RPCS2,%BDAOR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
      (2) 003236 060037 001122              ADD      RO,%BDAOR ;ADD RH11 BASE ADDRESS
      (2) 003242 005037 001124              CLR      %GDDAT ;WHAT REGISTER SHOULD BE
      (2) 003246 013737 001126 001164      MOV      %BDDAT,%TMP0 ;MOVE REGISTER CONTENTS TO '%TMP0'
      (2) 003254 042737 167777 001164      BIC      %1CNED,%TMP0 ;SAVE SPECIFIED BITS
      (2) 003262 023737 001124 001164      CMP      %GDDAT,%TMP0 ;COMPARE THE BITS
      (2) 003270 001414              BEQ      64$ ;BR IF OK
      (2) 003272 013737 001126 001174      MOV      %BDDAT,%TMP4 ;COPY 'BAD DATA'
      (2) 003300 042737 010000 001174      BIC      %NED,%TMP4 ;CLEAR THE MASKED BITS
      (2) 003306 053737 001174 001124      BIS      %TMP4,%GDDAT ;'OR' WITH GOOD DATA FOR TYPEOUT
      (2) 003314 104037              ERROR    37 ;TYPE MESSAGE 37
      (2) 003316 005137 001244              COM      CKERR ;SET THE REGISTER COMPARE ERROR INDICATOR
      (2) 003322 000240              NOP
      (1) 003324 005737 001244      64$: TST      CKERR ;WAS 'NED' SET ?
      (1) 003330 001403              BEQ      .+10 ;BR IF NOT
      (1) 003332 012760 000040 000010      MOV      %CLR,RPCS2(RO) ;ISSUE MASSBUS INIT TO CLEAR 'NED'
      (2) 003340 113760 001226 000010      MOV      PORTB,RPCS2(RO) ;SELECT PORT B
      (2) 003346 013737 001226 001234      MOV      PORTB,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
      (1) 003354 005760 000012              TST      RPS1(RO) ;SEE IF DRIVE (PORT B) PRESENT
      (2) 003360 005037 001244              CLR      CKERR ;CLEAR THE 'CHECK ERROR' INDICATOR
      (2) 003364 016037 000010 001126      MOV      RPCS2(RO),%BDDAT ;GET CONTENTS OF RPCS2
      (2) 003372 012737 000010 001122      MOV      %RPCS2,%BDAOR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
      (2) 003400 060037 001122              ADD      RO,%BDAOR ;ADD RH11 BASE ADDRESS
      (2) 003404 005037 001124              CLR      %GDDAT ;WHAT REGISTER SHOULD BE
      (2) 003410 013737 001126 001164      MOV      %BDDAT,%TMP0 ;MOVE REGISTER CONTENTS TO '%TMP0'
      (2) 003416 042737 167777 001164      BIC      %1CNED,%TMP0 ;SAVE SPECIFIED BITS
      (2) 003424 023737 001124 001164      CMP      %GDDAT,%TMP0 ;COMPARE THE BITS
      (2) 003432 001414              BEQ      66$ ;BR IF OK
      (2) 003434 013737 001126 001174      MOV      %BDDAT,%TMP4 ;COPY 'BAD DATA'
      (2) 003442 042737 010000 001174      BIC      %NED,%TMP4 ;CLEAR THE MASKED BITS
      (2) 003450 053737 001174 001124      BIS      %TMP4,%GDDAT ;'OR' WITH GOOD DATA FOR TYPEOUT
      (2) 003456 104037              ERROR    37 ;TYPE MESSAGE 37
      (2) 003460 005137 001244              COM      CKERR ;SET THE REGISTER COMPARE ERROR INDICATOR
      (2) 003464 000240              NOP
      (1) 003466 005737 001244      66$: TST      CKERR ;WAS 'NED' SET ?
      (1) 003472 001403              BEQ      .+10 ;BR IF NOT
      (1) 003474 012760 000040 000010      MOV      %CLR,RPCS2(RO) ;ISSUE MASSBUS INIT TO CLEAR 'NED'

```

;CONFIRM THAT DRIVE IS AN RP04/5/6 AND IS DUAL PORT

```

7955
7956
7957
7958
7962 003502 113760 001224 000010      MOV      PORTA,RPCS2(RO) ;SELECT PORT A
      (2) 003510 013737 001224 001234      MOV      PORTA,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
      (2) 003516 005037 001244              CLR      CKERR ;CLEAR THE 'CHECK ERROR' INDICATOR
      (2) 003522 016037 000026 001126      MOV      RPDT(RO),%BDDAT ;GET CONTENTS OF RPDT
      (2) 003530 012737 000026 001122      MOV      %RPDT,%BDAOR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
      (2) 003536 060037 001122              ADD      RO,%BDAOR ;ADD RH11 BASE ADDRESS
      (2) 003542 012737 024020 001124      MOV      %24020,%GDDAT ;WHAT REGISTER SHOULD BE
      (2) 003550 013737 001126 001164      MOV      %BDDAT,%TMP0 ;MOVE REGISTER CONTENTS TO '%TMP0'

```

```

(2) 003556 042737 000003 001164 BIC #1C177774,$TMP0 ;SAVE SPECIFIED BITS
(2) 003564 023737 001124 001164 CMP $GDDAT,$TMP0 ;COMPARE THE BITS
(2) 003572 001414 BEQ 68$ ;BR IF OK
(2) 003574 013737 001126 001174 MOV $BDDAT,$TMP4 ;COPY 'BAD DATA'
(2) 003602 042737 177774 001174 BIC #177774,$TMP4 ;CLEAR THE MASKED BITS
(2) 003610 053737 001174 001124 BIS $TMP4,$GDDAT ;'OR' WITH GOOD DATA FOR TYPEOUT
(2) 003616 104001 ERROR 1 ;TYPE MESSAGE 1
(2) 003620 005137 001244 COM CKERR ;SET THE REGISTER COMPARE ERROR INDICATOR
(2) 003624 000240 68$: NOP
(2) 003626 113760 001226 000010 MOVB PORTB,RPCS2(RO) ;SELECT PORT B
(2) 003634 013737 001226 001234 MOV PORTB,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
(2) 003642 005037 001244 CLR CKERR ;CLEAR THE 'CHECK ERROR' INDICATOR
(2) 003646 016037 000026 001126 MOV RPDT(RO),$BDDAT ;GET CONTENTS OF RPDT
(2) 003654 012737 000026 001122 MOV #RPDT,$B0ADR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
(2) 003662 060037 001122 ADD RO,$B0ADR ;ADD RH11 BASE ADDRESS
(2) 003666 012737 024020 001124 MOV #24020,$GDDAT ;WHAT REGISTER SHOULD BE
(2) 003674 013737 001126 001164 MOV $BDDAT,$TMP0 ;MOVE REGISTER CONTENTS TO '$TMP0'
(2) 003702 042737 000003 001164 BIC #1C177774,$TMP0 ;SAVE SPECIFIED BITS
(2) 003710 023737 001124 001164 CMP $GDDAT,$TMP0 ;COMPARE THE BITS
(2) 003716 001414 BEQ 70$ ;BR IF OK
(2) 003720 013737 001126 001174 MOV $BDDAT,$TMP4 ;COPY 'BAD DATA'
(2) 003726 042737 177774 001174 BIC #177774,$TMP4 ;CLEAR THE MASKED BITS
(2) 003734 053737 001174 001124 BIS $TMP4,$GDDAT ;'OR' WITH GOOD DATA FOR TYPEOUT
(2) 003742 104001 ERROR 1 ;TYPE MESSAGE 1
(2) 003744 005137 001244 COM CKERR ;SET THE REGISTER COMPARE ERROR INDICATOR
(2) 003750 000240 70$: NOP

```

```

7963
7964 ;*****
7965 ;VERIFY THROUGH BOTH PORTS THAT THE DRIVE IS ON LINE AND IN NEUTRAL
7966

```

```

7971 003752 113760 001224 000010 MOVB PORTA,RPCS2(RO) ;SELECT PORT A
(2) 003760 013737 001224 001234 MOV PORTA,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
(2) 003766 005037 001244 CLR CKERR ;CLEAR THE 'CHECK ERROR' INDICATOR
(2) 003772 016037 000012 001126 MOV RPDS1(RO),$BDDAT ;GET CONTENTS OF RPDS1
(2) 004000 012737 000012 001122 MOV #RPDS1,$B0ADR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
(2) 004006 060037 001122 ADD RO,$B0ADR ;ADD RH11 BASE ADDRESS
(2) 004012 012737 001000 001124 MOV #PGM,$GDDAT ;WHAT REGISTER SHOULD BE
(2) 004020 013737 001126 001164 MOV $BDDAT,$TMP0 ;MOVE REGISTER CONTENTS TO '$TMP0'
(2) 004026 042737 176777 001164 BIC #1CPGM,$TMP0 ;SAVE SPECIFIED BITS
(2) 004034 023737 001124 001164 CMP $GDDAT,$TMP0 ;COMPARE THE BITS
(2) 004042 001414 BEQ 72$ ;BR IF OK
(2) 004044 013737 001126 001174 MOV $BDDAT,$TMP4 ;COPY 'BAD DATA'
(2) 004052 042737 001000 001174 BIC #PGM,$TMP4 ;CLEAR THE MASKED BITS
(2) 004060 053737 001174 001124 BIS $TMP4,$GDDAT ;'OR' WITH GOOD DATA FOR TYPEOUT
(2) 004066 104045 ERROR 45 ;TYPE MESSAGE 45
(2) 004070 005137 001244 COM CKERR ;SET THE REGISTER COMPARE ERROR INDICATOR
(2) 004074 000240 72$: NOP
(2) 004076 005037 001244 CLR CKERR ;CLEAR THE 'CHECK ERROR' INDICATOR
(2) 004102 016037 000012 001126 MOV RPDS1(RO),$BDDAT ;GET CONTENTS OF RPDS1
(2) 004110 012737 000012 001122 MOV #RPDS1,$B0ADR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
(2) 004116 060037 001122 ADD RO,$B0ADR ;ADD RH11 BASE ADDRESS
(2) 004122 012737 010600 001124 MOV #MOL!DPR!DRY,$GDDAT ;WHAT REGISTER SHOULD BE
(2) 004130 013737 001126 001164 MOV $BDDAT,$TMP0 ;MOVE REGISTER CONTENTS TO '$TMP0'
(2) 004136 042737 167177 001164 BIC #1C10600,$TMP0 ;SAVE SPECIFIED BITS
(2) 004144 023737 001124 001164 CMP $GDDAT,$TMP0 ;COMPARE THE BITS
(2) 004152 001414 BEQ 74$ ;BR IF OK

```

```

(2) 004154 013737 001126 001174 MOV $BDDAT,$TMP4 ;COPY 'BAD DATA'
(2) 004162 042737 010600 001174 BIC #10600,$TMP4 ;CLEAR THE MASKED BITS
(2) 004170 053737 001174 001124 BIS $TMP4,$GDDAT ;'OR' WITH GOOD DATA FOR TYPEOUT
(2) 004176 104002 ERROR 2 ;TYPE MESSAGE 2
(2) 004200 005137 001244 COM CKERR ;SET THE REGISTER COMPARE ERROR INDICATOR
(2) 004204 000240 74$: NOP
(2) 004206 113760 001226 000010 MOVB PORTB,RPCS2(RO) ;SELECT PORT B
(2) 004214 013737 001226 001234 MOV PORTB,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
(2) 004222 005037 001244 CLR CKERR ;CLEAR THE 'CHECK ERROR' INDICATOR
(2) 004226 016037 000012 001126 MOV RPDS1(RO),$BDDAT ;GET CONTENTS OF RPDS1
(2) 004234 012737 000012 001122 MOV #RPS1,$BDAOR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
(2) 004242 060037 001122 ADD RO,$BDAOR ;ADD RH11 BASE ADDRESS
(2) 004246 012737 001000 001124 MOV #PGM,$GDDAT ;WHAT REGISTER SHOULD BE
(2) 004254 013737 001126 001164 MOV $BDDAT,$TMP0 ;MOVE REGISTER CONTENTS TO '$TMP0'
(2) 004262 042737 176777 001164 BIC #1CPGM,$TMP0 ;SAVE SPECIFIED BITS
(2) 004270 023737 001124 001164 CMP $GDDAT,$TMP0 ;COMPARE THE BITS
(2) 004276 001414 BEQ 76$ ;BR IF OK
(2) 004300 013737 001126 001174 MOV $BDDAT,$TMP4 ;COPY 'BAD DATA'
(2) 004306 042737 001000 001174 BIC #PGM,$TMP4 ;CLEAR THE MASKED BITS
(2) 004314 053737 001174 001124 BIS $TMP4,$GDDAT ;'OR' WITH GOOD DATA FOR TYPEOUT
(2) 004322 104045 ERROR 45 ;TYPE MESSAGE 45
(2) 004324 005137 001244 COM CKERR ;SET THE REGISTER COMPARE ERROR INDICATOR
(2) 004330 000240 76$: NOP
(2) 004332 005037 001244 CLR CKERR ;CLEAR THE 'CHECK ERROR' INDICATOR
(2) 004336 016037 000012 001126 MOV RPDS1(RO),$BDDAT ;GET CONTENTS OF RPDS1
(2) 004344 012737 000012 001122 MOV #RPS1,$BDAOR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
(2) 004352 060037 001122 ADD RO,$BDAOR ;ADD RH11 BASE ADDRESS
(2) 004356 012737 010600 001124 MOV #MOL!DPR!DRY,$GDDAT ;WHAT REGISTER SHOULD BE
(2) 004364 013737 001126 001164 MOV $BDDAT,$TMP0 ;MOVE REGISTER CONTENTS TO '$TMP0'
(2) 004372 042737 167177 001164 BIC #1C10600,$TMP0 ;SAVE SPECIFIED BITS
(2) 004400 023737 001124 001164 CMP $GDDAT,$TMP0 ;COMPARE THE BITS
(2) 004406 001414 BEQ 78$ ;BR IF OK
(2) 004410 013737 001126 001174 MOV $BDDAT,$TMP4 ;COPY 'BAD DATA'
(2) 004416 042737 010600 001174 BIC #10600,$TMP4 ;CLEAR THE MASKED BITS
(2) 004424 053737 001174 001124 BIS $TMP4,$GDDAT ;'OR' WITH GOOD DATA FOR TYPEOUT
(2) 004432 104002 ERROR 2 ;TYPE MESSAGE 2
(2) 004434 005137 001244 COM CKERR ;SET THE REGISTER COMPARE ERROR INDICATOR
(2) 004440 000240 78$: NOP

```

```

7972
7973 ;:*****
7974 ;VERIFY THAT DRIVE SERIAL NUMBER SEEN THROUGH BOTH PORTS IS THE SAME
7975

```

```

7976 004442 113760 001224 000010 MOVB PORTA,RPCS2(RO) ;SELECT PORT A
7977 004450 016037 000030 001124 MOV RPSN(RO),$GDDAT ;STORE THE PORT A SERIAL NUMBER
7978 004456 113760 001226 000010 MOVB PORTB,RPCS2(RO) ;SELECT PORT B
7979 004464 016037 000030 001126 MOV RPSN(RO),$BDDAT ;STORE THE PORT B SERIAL NUMBER
7980 004472 023737 001124 001126 CMP $GDDAT,$BDDAT ;ARE THEY THE SAME?
7981 004500 001406 BEQ 1$ ;BR IF THEY ARE
7982 004502 104003 ERROR 3 ;REPORT THE ERROR
7983 004504 032777 100000 174426 BIT #SW15,$SWR ;HALT ON ERROR?
7984 004512 001001 BNE 1$ ;BR IF SET - PROGRAM HAS ALREADY HALTED
7985 004514 000000 HALT ;HALT, POSSIBLE CABLE CONNECTION PROBLEM
7986 004516 000004 1$: SCOPE ;LOOP?
7987
8005
8006

```

```

;:*****

```



```

(2) 004732 000137 006116          JMP      55          ;BYPASS REST OF THE SUBTEST
(2) 004736          64$:          MOV      PORTA,RPCS2(RO) ;SELECT PORT A
(3) 004736 113760 001224 000010      MOV      PORTA,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
(3) 004744 013737 001224 001234      MOV      RPD51(RO),SDDAT ;SEE IF SEIZING PORT SEES CORRECT STATUS
(2) 004752 016037 000012 001126      MOV      #MOL:PGM:OPR:DRY,SDDAT ;EXPECTED STATUS
(2) 004760 012737 011600 001124      MOV      $SGDAT,$STMP1 ;USE GOOD DATA AS A MASK
(2) 004766 013737 001124 001166      MOV      $SGDAT,$STMP1 ;COMPLEMENT THE EXPECTED STATUS
(2) 004774 005137 001166          COM          ;SAVE THE ACTUAL STATUS
(2) 005000 013737 001126 001164      MOV      SDDAT,$STMP0
(2) 005006 043737 001166 001164      BIC      $STMP1,$STMP0 ;CLEAR UNWANTED BITS
(2) 005014 023737 001124 001164      CMP      $SGDAT,$STMP0 ;ARE THE EXPECTED STATUS BITS SET
(2) 005022 001401          BEQ      65$        ;BR IF THEY ARE
(2) 005024 104005          ERROR 5 ;REPORT THE ERROR
(2) 005026 000240          65$:          NOP

```

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

```

(2) 005030 113760 001226 000010      MOV      PORTB,RPCS2(RO) ;SELECT PORT B
(2) 005036 013737 001226 001234      MOV      PORTB,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
(2) 005044 016046 000046          MOV      RPEC2(RO),-(SP) ;STORE REGISTER RPEC2, PORT B, FOR CHECK
(2) 005050 016046 000044          MOV      RPEC1(RO),-(SP) ;STORE REGISTER RPEC1, PORT B, FOR CHECK
(2) 005054 016046 000042          MOV      RPER3(RO),-(SP) ;STORE REGISTER RPER3, PORT B, FOR CHECK
(2) 005060 016046 000030          MOV      RPSN(RO),-(SP) ;STORE REGISTER RPSN, PORT B, FOR CHECK
(2) 005064 016046 000036          MOV      RPCC(RO),-(SP) ;STORE REGISTER RPCC, PORT B, FOR CHECK
(2) 005070 016046 000034          MOV      RPCA(RO),-(SP) ;STORE REGISTER RPCA, PORT B, FOR CHECK
(2) 005074 016046 000032          MOV      RPOF(RO),-(SP) ;STORE REGISTER RPOF, PORT B, FOR CHECK
(2) 005100 016046 000040          MOV      RPER2(RO),-(SP) ;STORE REGISTER RPER2, PORT B, FOR CHECK
(2) 005104 016046 000020          MOV      RPLA(RO),-(SP) ;STORE REGISTER RPLA, PORT B, FOR CHECK
(2) 005110 016046 000026          MOV      RPD1(RO),-(SP) ;STORE REGISTER RPD1, PORT B, FOR CHECK
(2) 005114 016046 000006          MOV      RPDA(RO),-(SP) ;STORE REGISTER RPDA, PORT B, FOR CHECK
(2) 005120 016046 000024          MOV      RPMR(RO),-(SP) ;STORE REGISTER RPMR, PORT B, FOR CHECK
(2) 005124 016046 000014          MOV      RPER1(RO),-(SP) ;STORE REGISTER RPER1, PORT B, FOR CHECK

```

:WAIT FOR PORT A TO TIMEOUT

```

(1) 005130 005760 000012          1$:          TST      RPD51(RO) ;WAIT FOR THE DRIVE TO TIMEOUT
(1) 005134 001006          BNE      2$          ;BR WHEN TIMEOUT OCCURS
(1) 005136 005737 001254          TST      WATCH ;CHECK WATCH
(1) 005142 001372          BNE      1$          ;BR IF NOT ZERO
(1) 005144 104036          ERROR 36 ;NO TIMEOUT WITHIN 2 SECONDS
(1) 005146 000137 005536          JMP      4$          ;BYPASS TIMEOUT TIME CHECK
(1) 005152 012737 000340 177776          2$:          MOV      #(<7*32.)>,$#PS ;SET PRIORITY TO 7 TO STOP CLOCK
(1) 005160 013737 001252 001256          MOV      TIME,TIMEA ;SAVE THE ELAPSED TIME FOR PORT A
(1) 005166 004537 056660          JSR      RS,TOLER ;CALCULATE THE TOLERANCE
(1) 005172 001256          .WORD   TIMEA ;TIMEOUT VALUE FOR PORT A
(1) 005174 012637 001260          MOV      (SP)+,TIMEAP ;+25% TOLERANCE
(1) 005200 012637 001262          MOV      (SP)+,TIMEAM ;-25% TOLERANCE

```

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

```

(1) 005204 023727 001252 000764          CMP      TIME,#500. ;WAS MEASURED TIME AT LEAST 500 MS?
(1) 005212 103001          BHS      3$          ;BR IF IT WAS
(1) 005214 104055          ERROR 55 ;REPORT TIMEOUT TOO SHORT

```

```

(1)
(2)
(1)
(1) 005216 012737 000240 177776 35: MOV #(<S*32.),20PS ;RESTORE PRIORITY TO 5
(2)
(2) ;VERIFY THA: THE DRIVE IS IN NEUTRAL
(2)
(2) 005224 005037 001250 CLR RELERR ;CLEAR THE 'RELEASE ERROR' INDICATOR
(2) 005230 012737 000012 001122 MOV #RPDS1,$BDAOR ;FORM THE ADDRESS OF RPDS1 FOR TYPEOUT
(2) 005236 060037 001122 ADD RO,$BDAOR ;ADD THE I/O BASE ADDRESS
(2) 005242 012737 011600 001124 MOV #MOL!PGM!DPR!DRY,$GDDAT ;COMPARISON CONSTANT
(2) 005250 113760 001224 000010 MOVB PORTA,RPCS2(RO) ;SELECT PORT A.
(2) 005256 016037 000012 001170 MOV RPDS1(RO),$TMP2 ;GET THE DRIVE STATUS REGISTER FROM PORT A.
(2) 005264 013737 001170 001164 MOV $TMP2,$TMP0 ;COPY IT INTO '$TMP0'
(2) 005272 042737 100100 001164 BIC #ATA!VV,$TMP0 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
(2) 005300 113760 001226 000010 MOVB PORTB,RPCS2(RO) ;SELECT PORT B.
(2) 005306 016037 000012 001172 MOV RPDS1(RO),$TMP3 ;GET THE DRIVE STATUS REGISTER FROM PORT B.
(2) 005314 013737 001172 001166 MOV $TMP3,$TMP1 ;COPY IT INTO '$TMP1'
(2) 005322 042737 100100 001166 BIC #ATA!VV,$TMP1 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
(2) 005330 023737 001164 001166 CMP $TMP0,$TMP1 ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
(2) 005336 001006 BNE 66$ ;BR IF NOT
(2) 005340 005737 001164 TST $TMP0 ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
(2) 005344 001037 BNE 68$ ;BR IF NOT
(2) 005346 104046 ERROR 46 ;REPORT DRIVE NOT IN NEUTRAL OR NOT SEIZED
(2) 005350 000137 005534 JMP 70$ ;BYPASS THE REST OF THE CHECKS
(2) 005354 013737 001170 001126 66$: MOV $TMP2,$BDDAT ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
(2) 005362 013737 001226 001234 MOV PORTB,PTNBR ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
(2) 005370 113760 001226 000010 MOVB PORTB,RPCS2(RO) ;SELECT PORT B.
(2) 005376 005737 001164 TST $TMP0 ;SEE IF STATUS EQ 0 FROM PORT A.
(2) 005402 001414 BEQ 67$ ;BR IF ZERO
(2) 005404 013737 001224 001234 MOV PORTA,PTNBR ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
(2) 005412 013737 001172 001126 MOV $TMP3,$BDDAT ;'BAD DATA' FOR ERROR TYPE OUT
(2) 005420 113760 001224 000010 MOVB PORTA,RPCS2(RO) ;SELECT PORT A.
(2) 005426 005737 001166 TST $TMP1 ;SEE IF STATUS EQ ZERO FROM PORT B.
(2) 005432 001004 BNE 68$ ;BR IF NOT
(2) 005434 012737 177777 001250 67$: MOV #-1,RELERR ;SET 'RELEASE ERROR' INDICATOR
(2) 005442 104022 ERROR 22 ;TYPE ERROR MESSAGE 22
(2) 005444 013737 001170 001126 68$: MOV $TMP2,$BDDAT ;LOOK FOR BIT FAILURES WHEN RPDS1 READ
(2) 005452 013737 001224 001234 MOV PORTA,PTNBR ;CHANGE PORT NUMBER
(2) 005460 042737 100100 001170 BIC #ATA!VV,$TMP2 ;DON'T CHECK ATTN BIT OR VV BIT
(2) 005466 023737 001124 001170 CMP $GDDAT,$TMP2 ;ALL BITS OK ?
(2) 005474 001401 BEQ 69$ ;BR IF OK FROM PORT A.
(2) 005476 104007 ERROR 7 ;REPORT ERROR
(2) 005500 013737 001172 001126 69$: MOV $TMP3,$BDDAT ;CHECK RPDS1 FOR BIT FAILURES - FROM PORT B.
(2) 005506 013737 001226 001234 MOV PORTB,PTNBR ;CHANGE PORT NUMBER
(2) 005514 042737 100100 001172 BIC #ATA!VV,$TMP3 ;DON'T CHECK ATTN BIT OR VV BIT
(2) 005522 023737 001124 001172 CMP $GDDAT,$TMP3 ;SEE IF READ OK FROM PORT B.
(2) 005530 001401 BEQ 70$ ;BR IF OK
(2) 005532 104007 ERROR 7 ;REPORT ERROR
(2) 005534 000240 70$: NOP

```

```

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

```

(1)	005536	013737	001226	001234	45:	MOV	PORTB,PTNBR	:CHANGE 'PORT NUMBER' TO THE OPPOSITE PORT
(2)	005544	010037	001122			MOV	RO,\$BDADR	:BASE ADDRESS FOR REGISTER RPER1
(2)	005550	062737	000014	001122		ADD	#RPER1,\$BDADR	:ADDRESS OF RPER1 FOR TYPEOUT
(2)	005556	012637	001126			MOV	(SP)+,\$BDDAT	:CHECK THE STORED CONTENTS OF RPER1
(2)	005562	001401				BEQ	.+4	:CONTENTS ZERO ?
(2)	005564	104006				ERROR	6	:REPORT THAT PORT B SAW NON-ZERO REGISTER
(2)	005566	010037	001122			MOV	RO,\$BDADR	:BASE ADDRESS FOR REGISTER RPMR
(2)	005572	062737	000024	001122		ADD	#RPMR,\$BDADR	:ADDRESS OF RPMR FOR TYPEOUT
(2)	005600	012637	001126			MOV	(SP)+,\$BDDAT	:CHECK THE STORED CONTENTS OF RPMR
(2)	005604	001401				BEQ	.+4	:CONTENTS ZERO ?
(2)	005606	104006				ERROR	6	:REPORT THAT PORT B SAW NON-ZERO REGISTER
(2)	005610	010037	001122			MOV	RO,\$BDADR	:BASE ADDRESS FOR REGISTER RPDA
(2)	005614	062737	000006	001122		ADD	#RPDA,\$BDADR	:ADDRESS OF RPDA FOR TYPEOUT
(2)	005622	012637	001126			MOV	(SP)+,\$BDDAT	:CHECK THE STORED CONTENTS OF RPDA
(2)	005626	001401				BEQ	.+4	:CONTENTS ZERO ?
(2)	005630	104006				ERROR	6	:REPORT THAT PORT B SAW NON-ZERO REGISTER
(2)	005632	010037	001122			MOV	RO,\$BDADR	:BASE ADDRESS FOR REGISTER RPDT
(2)	005636	062737	000026	001122		ADD	#RPDT,\$BDADR	:ADDRESS OF RPDT FOR TYPEOUT
(2)	005644	012637	001126			MOV	(SP)+,\$BDDAT	:CHECK THE STORED CONTENTS OF RPDT
(2)	005650	001401				BEQ	.+4	:CONTENTS ZERO ?
(2)	005652	104006				ERROR	6	:REPORT THAT PORT B SAW NON-ZERO REGISTER
(2)	005654	010037	001122			MOV	RO,\$BDADR	:BASE ADDRESS FOR REGISTER RPLA
(2)	005660	062737	000020	001122		ADD	#RPLA,\$BDADR	:ADDRESS OF RPLA FOR TYPEOUT
(2)	005666	012637	001126			MOV	(SP)+,\$BDDAT	:CHECK THE STORED CONTENTS OF RPLA
(2)	005672	001401				BEQ	.+4	:CONTENTS ZERO ?
(2)	005674	104006				ERROR	6	:REPORT THAT PORT B SAW NON-ZERO REGISTER
(2)	005676	010037	001122			MOV	RO,\$BDADR	:BASE ADDRESS FOR REGISTER RPER2
(2)	005702	062737	000040	001122		ADD	#RPER2,\$BDADR	:ADDRESS OF RPER2 FOR TYPEOUT
(2)	005710	012637	001126			MOV	(SP)+,\$BDDAT	:CHECK THE STORED CONTENTS OF RPER2
(2)	005714	001401				BEQ	.+4	:CONTENTS ZERO ?
(2)	005716	104006				ERROR	6	:REPORT THAT PORT B SAW NON-ZERO REGISTER
(2)	005720	010037	001122			MOV	RO,\$BDADR	:BASE ADDRESS FOR REGISTER RPOF
(2)	005724	062737	000032	001122		ADD	#RPOF,\$BDADR	:ADDRESS OF RPOF FOR TYPEOUT
(2)	005732	012637	001126			MOV	(SP)+,\$BDDAT	:CHECK THE STORED CONTENTS OF RPOF
(2)	005736	001401				BEQ	.+4	:CONTENTS ZERO ?
(2)	005740	104006				ERROR	6	:REPORT THAT PORT B SAW NON-ZERO REGISTER
(2)	005742	010037	001122			MOV	RO,\$BDADR	:BASE ADDRESS FOR REGISTER RPCA
(2)	005746	062737	000034	001122		ADD	#RPCA,\$BDADR	:ADDRESS OF RPCA FOR TYPEOUT
(2)	005754	012637	001126			MOV	(SP)+,\$BDDAT	:CHECK THE STORED CONTENTS OF RPCA
(2)	005760	001401				BEQ	.+4	:CONTENTS ZERO ?
(2)	005762	104006				ERROR	6	:REPORT THAT PORT B SEES NON-ZERO REGISTER
(2)	005764	010037	001122			MOV	RO,\$BDADR	:BASE ADDRESS FOR REGISTER RPCC
(2)	005770	062737	000036	001122		ADD	#RPCC,\$BDADR	:ADDRESS OF RPCC FOR TYPEOUT
(2)	005776	012637	001126			MOV	(SP)+,\$BDDAT	:CHECK THE STORED CONTENTS OF RPCC
(2)	006002	001401				BEQ	.+4	:CONTENTS ZERO ?
(2)	006004	104006				ERROR	6	:REPORT THAT PORT B SEES NON-ZERO REGISTER
(2)	006006	010037	001122			MOV	RO,\$BDADR	:BASE ADDRESS FOR REGISTER RPSN
(2)	006012	062737	000030	001122		ADD	#RPSN,\$BDADR	:ADDRESS OF RPSN FOR TYPEOUT
(2)	006020	012637	001126			MOV	(SP)+,\$BDDAT	:CHECK THE STORED CONTENTS OF RPSN
(2)	006024	001401				BEQ	.+4	:CONTENTS ZERO ?
(2)	006026	104006				ERROR	6	:REPORT THAT PORT B SEES NON-ZERO REGISTER
(2)	006030	010037	001122			MOV	RO,\$BDADR	:BASE ADDRESS FOR REGISTER RPER3
(2)	006034	062737	000042	001122		ADD	#RPER3,\$BDADR	:ADDRESS OF RPER3 FOR TYPEOUT
(2)	006042	012637	001126			MOV	(SP)+,\$BDDAT	:CHECK THE STORED CONTENTS OF RPER3
(2)	006046	001401				BEQ	.+4	:CONTENTS ZERO ?
(2)	006050	104006				ERROR	6	:REPORT THAT PORT B SEES NON-ZERO REGISTER


```

(2) 006052 010037 001122      MOV      R0,$BDADR      ;BASE ADDRESS FOR REGISTER RPEC1
(2) 006056 062737 000044 001122  ADD      #RPEC1,$BDADR  ;ADDRESS OF RPEC1 FOR TIMEOUT
(2) 006064 012637 001126      MOV      (SP)+,$BDADR   ;CHECK THE STORED CONTENTS OF RPEC1
(2) 006070 001401                BEQ      .+4            ;CONTENTS ZERO ?
(2) 006072 104006                ERROR    6              ;REPORT THAT PORT B SEES NON-ZERO REGISTER
(2) 006074 010037 001122      MOV      R0,$BDADR      ;BASE ADDRESS FOR REGISTER RPEC2
(2) 006100 062737 000046 001122  ADD      #RPEC2,$BDADR  ;ADDRESS OF RPEC2 FOR TIMEOUT
(2) 006106 012637 001126      MOV      (SP)+,$BDADR   ;CHECK THE STORED CONTENTS OF RPEC2
(2) 006112 001401                BEQ      .+4            ;CONTENTS ZERO ?
(2) 006114 104006                ERROR    6              ;REPORT THAT PORT B SEES NON-ZERO REGISTER
(1) 006116 000004                S$:      SCOPE          ;LOOP ?

```

```

(3) *****
(3) *TEST 3          PORT 'B' SEIZE/TIMEOUT TEST
(3) *
(3) *VERIFY THAT THE DRIVE CAN BE SEIZED AND THAT THE PORT TIMEOUT RELEASES
(3) *THE DRIVE.
(3) *
(3) * A. WRITE 0'S INTO RPD51 THROUGH PORT 'B'; VERIFY THAT THE DRIVE
(3) *HAS BEEN SEIZED.
(3) *
(3) * B. READ EACH DRIVE REGISTER, EXCEPT RPCS1, THROUGH PORT 'A';
(3) *VERIFY THAT 0'S ARE READ FROM EACH REGISTER.
(3) *
(3) * C. WAIT FOR THE PORT TIMEOUT TO OCCUR AND RELEASE THE DRIVE.
(3) *MEASURE THE DURATION OF THE TIMEOUT ONE SHOT AND SAVE THE
(3) *VALUE FOR LATER USE. VERIFY THAT TIMEOUT RETURNED THE DRIVE TO
(3) *NEUTRAL.
(3) *****

```

```

(3) 006120
(3) 006120 005737 001274      TST      KYBCTL         ;PERFORMING ONLY SINGLE TESTS ?
(3) 006124 001406                BEQ      2$            ;BR IF NOT
(3) 006126 100002                BPL      1$            ;BR IF JUST ENTERED TEST
(3) 006130 000137 002622      JMP      EXEC           ;RETURN & GET NEXT TEST NUMBER
(3) 006134 012737 177777 001274 1$:      MOV      #-1,KYBCTL     ;SET SINGLE TEST INDICATOR
(3) 006142 112737 000003 001102 2$:      MOV      #3,$STNM      ;TEST NUMBER
(3) 006150 012737 006172 001106      MOV      #TEST3,$LPADR ;LOAD LOOP ON TEST ADDRESS
(3) 006156 012737 006172 001110      MOV      #TEST3,$LPERR ;LOAD LOOP ON ERROR ADDRESS
(1) 006164 012737 000012 001176      MOV      #10,$TIMES    ;DO 10 ITERATIONS
(1) 006172 012706 001100      TEST3: MOV      #STACK,SP ;LOAD THE STACK POINTER
(1) 006176 012737 000240 177776      MOV      #<5*32>,$#PS  ;SET PRIORITY TO 5 IN CASE LOOPING
(1) 006204 005037 001264      CLR      TIMEB         ;CLEAR TIMEOUT VALUE FOR PORT B
(1) 006210 005037 001266      CLR      TIMEBP        ;CLEAR UPPER TIMEOUT TOLERANCE
(1) 006214 005037 001270      CLR      TIMEBM        ;CLEAR LOWER TIMEOUT TOLERANCE

```

```

(3) *****
(3) ;START THE TIMER
(3) *****
(3) 006220 005037 001252      CLR      TIME          ;CLEAR THE ELAPSED TIME COUNTER
(3) 006224 012737 003720 001254      MOV      #2000.,WATCH ;SET WATCH TO 2000 MS
(3) *****
(3) ;SEIZE THE DRIVE THROUGH PORT B

```

```

(2) 006232 113760 001226 000010   MOVB   PORTB,RPCS2(RO) ;SELECT PORT B
(2) 006240 013737 001226 001236   MOV    PORTB,SEIZPT ;STORE SEIZING PORT'S ADDRESS
(2) 006246 005060 000012         CLR    RPDS1(RO) ;WRITE RPDS1
(3) 006252 113760 001224 000010   MOVB   PORTA,RPCS2(RO) ;SELECT PORT A
(3) 006260 013737 001224 001234   MOV    PORTA,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
(2) 006266 013737 001224 001240   MOV    PORTA,OPPRT ;'OPPOSITE' PORT ADDRESS
(2) 006274 016037 000012 001126   MOV    RPDS1(RO),SBDDAT ;SEE IF DRIVE SEIZED BY PORT B
(2) 006302 010037 001122         MOV    RO,$BDAADR ;RHL1 BASE ADDRESS
(2) 006306 062737 000012 001122   ADD    #RPDS1,$BDAADR ;GENERATE BAD REGISTER ADDRESS
(2) 006314 005037 001124         CLR    $GDDAT ;REGISTER SHOULD BE ZERO
(2) 006320 023737 001124 001126   CMP    $GDDAT,$BDDAT ;IS THE REGISTER ZERO
(2) 006326 001403         BEQ   64$ ;BR IF IT IS
(2) 006330 104004         ERROR 4 ;REPORT THE ERROR
(2) 006332 000137 007516         JMP   5$ ;BYPASS REST OF THE SUBTEST
(2) 006336         64$:
(3) 006336 113760 001226 000010   MOVB   PORTB,RPCS2(RO) ;SELECT PORT B
(3) 006344 013737 001226 001234   MOV    PORTB,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
(2) 006352 016037 000012 001126   MOV    RPDS1(RO),SBDDAT ;SEE IF SEIZING PORT SEES CORRECT STATUS
(2) 006360 012737 011600 001124   MOV    #MOL!PGM!OPR!DRY,$GDDAT ;EXPECTED STATUS
(2) 006366 013737 001124 001166   MOV    $GDDAT,$TMP1 ;USE GOOD DATA AS A MASK
(2) 006374 005137 001166         COM   $TMP1 ;COMPLEMENT THE EXPECTED STATUS
(2) 006400 013737 001126 001164   MOV    $BDDAT,$TMP0 ;SAVE THE ACTUAL STATUS
(2) 006406 043737 001166 001164   BIC   $TMP1,$TMP0 ;CLEAR UNWANTED BITS
(2) 006414 023737 001124 001164   CMP    $GDDAT,$TMP0 ;ARE THE EXPECTED STATUS BITS SET ?
(2) 006422 001401         BEQ   65$ ;BR IF THEY ARE
(2) 006424 104005         ERROR 5 ;REPORT THE ERROR
(2) 006426 000240         65$:
NOP

```

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

```

(2) 006430 113760 001224 000010   MOVB   PORTA,RPCS2(RO) ;SELECT PORT A
(2) 006436 013737 001224 001234   MOV    PORTA,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
(2) 006444 016046 000046         MOV    RPEC2(RO),-(SP) ;STORE REGISTER RPEC2, PORT A, FOR CHECK
(2) 006450 016046 000044         MOV    RPEC1(RO),-(SP) ;STORE REGISTER RPEC1, PORT A, FOR CHECK
(2) 006454 016046 000042         MOV    RPER3(RO),-(SP) ;STORE REGISTER RPER3, PORT A, FOR CHECK
(2) 006460 016046 000030         MOV    RPSN(RO),-(SP) ;STORE REGISTER RPSN, PORT A, FOR CHECK
(2) 006464 016046 000036         MOV    RPCC(RO),-(SP) ;STORE REGISTER RPCC, PORT A, FOR CHECK
(2) 006470 016046 000034         MOV    RPCA(RO),-(SP) ;STORE REGISTER RPCA, PORT A, FOR CHECK
(2) 006474 016046 000032         MOV    RPOF(RO),-(SP) ;STORE REGISTER RPOF, PORT A, FOR CHECK
(2) 006500 016046 000040         MOV    RPER2(RO),-(SP) ;STORE REGISTER RPER2, PORT A, FOR CHECK
(2) 006504 016046 000020         MOV    RPLA(RO),-(SP) ;STORE REGISTER RPLA, PORT A, FOR CHECK
(2) 006510 016046 000026         MOV    RPDT(RO),-(SP) ;STORE REGISTER RPDT, PORT A, FOR CHECK
(2) 006514 016046 000006         MOV    RPDA(RO),-(SP) ;STORE REGISTER RPDA, PORT A, FOR CHECK
(2) 006520 016046 000024         MOV    RPMR(RO),-(SP) ;STORE REGISTER RPMR, PORT A, FOR CHECK
(2) 006524 016046 000014         MOV    RPER1(RO),-(SP) ;STORE REGISTER RPER1, PORT A, FOR CHECK

```

:WAIT FOR PORT B TO TIMEOUT

```

(1) 006530 005760 000012 1$:   TST    RPDS1(RO) ;WAIT FOR THE DRIVE TO TIMEOUT
(1) 006534 001006         BNE   2$ ;BR WHEN TIMEOUT OCCURS
(1) 006536 005737 001254         TST    WATCH ;CHECK WATCH
(1) 006542 001372         BNE   1$ ;BR IF NOT ZERO
(1) 006544 104036         ERROR 36 ;NO TIMEOUT WITHIN 2 SECONDS
(1) 006546 000137 007136         JMP   4$ ;BYPASS TIMEOUT TIME CHECK

```

```

(1) 006552 012737 000340 177776 2$: MOV #(<7*32.>),2#PS ;SET PRIORITY TO 7 TO STOP CLOCK
(1) 006560 013737 001252 001264 MOV TIME,TIMEB ;SAVE THE ELAPSED TIME FOR PORT B
(1) 006566 004537 056660 JSR RS,TOLEF ;CALCULATE THE TOLERANCE
(1) 006572 001264 .WORD TIMEB ;TIMEOUT VALUE FOR PORT B
(1) 006574 012637 001266 MOV (SP)+,TIMEBP ;+25% TOLERANCE
(1) 006600 012637 001270 MCV (SP)+,TIMEBM ; -25% TOLERANCE
(1)
(2) ;*****
(1) ;VERIFY THAT THE TIMEOUT ONE-SHOT IS AT LEAST 500 MS
(1) 006604 023727 001252 000764 CMP TIME,#500. ;WAS MEASURED TIME AT LEAST 500 MS?
(1) 006612 103001 BHS 3$ ;BR IF IT WAS
(1) 006614 104055 ERROR 5$ ;REPORT TIMEOUT TOO SHORT
(1)
(2) ;*****
(1) ;VERIFY THAT THE DRIVE RETURNED TO NEUTRAL AFTER PORT B TIME OUT
(1)
(2) 006616 012737 000240 177776 3$: MOV #(<5*32.>),2#PS ;RESTORE PRIORITY TO 5
(2) ;VERIFY THAT THE DRIVE IS IN NEUTRAL
(2)
(2) 006624 005037 001250 CLR RELERR ;CLEAR THE 'RELEASE ERROR' INDICATOR
(2) 006630 012737 000012 001122 MOV #RPDS1,$BDDADR ;FORM THE ADDRESS OF RPDS1 FOR TYPEOUT
(2) 006636 060037 001122 ADD RO,$BDDADR ;ADD THE I/O BASE ADDRESS
(2) 006642 012737 011600 001124 MOV #MOL!PGM!DPR!DRY,$GDDAT ;COMPARISON CONSTANT
(2) 006650 113760 001224 000010 MOVB PORTA,APCS2(RO) ;SELECT PORT A.
(2) 006656 016037 000012 001170 MOV RPDS1(RO),$TMP2 ;GET THE DRIVE STATUS REGISTER FROM PORT A.
(2) 006664 013737 001170 001164 MOV $TMP2,$TMP0 ;COPY IT INTO '$TMP0'
(2) 006672 042737 100100 001164 BIC #ATA!VV,$TMP0 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
(2) 006700 113760 001226 000010 MOVB PORTB,APCS2(RO) ;SELECT PORT B.
(2) 006706 016037 000012 001172 MOV RPDS1(RO),$TMP3 ;GET THE DRIVE STATUS REGISTER FROM PORT B.
(2) 006714 013737 001172 001166 MOV $TMP3,$TMP1 ;COPY IT INTO '$TMP1'
(2) 006722 042737 100100 001166 BIC #ATA!VV,$TMP1 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
(2) 006730 023737 001164 001166 CMP $TMP0,$TMP1 ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
(2) 006736 001006 BNE 66$ ;BR IF NOT
(2) 006740 005737 001164 TST $TMP0 ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
(2) 006744 001037 BNE 68$ ;BR IF NOT
(2) 006746 104046 ERROR 46 ;REPORT DRIVE NOT IN NEUTRAL OR NOT SEIZED
(2) 006750 000137 007134 JMP 70$ ;BYPASS THE REST OF THE CHECKS
(2) 006754 013737 001170 001126 66$: MOV $TMP2,$BDDAT ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
(2) 006762 013737 001226 001234 MOV PORTB,PTNBR ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
(2) 006770 113760 001226 000010 MOVB PORTB,APCS2(RO) ;SELECT PORT B.
(2) 006776 005737 001164 TST $TMP0 ;SEE IF STATUS EQ 0 FROM PORT A.
(2) 007002 001414 BEQ 67$ ;BR IF ZERO
(2) 007004 013737 001224 001234 MOV PORTA,PTNBR ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
(2) 007012 013737 001172 001126 MOV $TMP3,$BDDAT ;'BAD DATA' FOR ERROR TYPE OUT
(2) 007020 113760 001224 000010 MOVB PORTA,APCS2(RO) ;SELECT PORT A.
(2) 007026 005737 001166 TST $TMP1 ;SEE IF STATUS EQ ZERO FROM PORT B.
(2) 007032 001004 BNE 68$ ;BR IF NOT
(2) 007034 012737 177777 001250 67$: MOV #-1,RELERR ;SET 'RELEASE ERROR' INDICATOR
(2) 007042 104022 ERROR 22 ;TYPE ERROR MESSAGE 22
(2) 007044 013737 001170 001126 68$: MOV $TMP2,$BDDAT ;LOOK FOR BIT FAILURES WHEN RPDS1 READ
(2) 007052 013737 001224 001234 MOV PORTA,PTNBR ;CHANGE PORT NUMBER
(2) 007060 042737 100100 001170 BIC #ATA!VV,$TMP2 ;DON'T CHECK ATTN BIT OR VV BIT
(2) 007066 023737 001124 001170 CMP $GDDAT,$TMP2 ;ALL BITS OK ?
(2) 007074 001401 BEQ 69$ ;BR IF OK FROM PORT A.

```

```

(2) 007076 104007          ERROR 7          ;REPORT ERROR
(2) 007100 013737 001172 001126 695:  MOV  $TMP3,$BDDAT ;CHECK RPOSI FOR BIT FAILURES - FROM PORT B.
(2) 007106 013737 001226 001234      MOV  PORTB,PTNBR ;CHANGE PORT NUMBER
(2) 007114 042737 100100 001172      BIC  #ATA,VV,$TMP3 ;DON'T CHECK ATTN BIT OR VV BIT
(2) 007122 023737 001124 001172      CMP  $GDDAT,$TMP3 ;SEE IF READ OK FROM PORT B.
(2) 007130 001401          BEQ  705          ;BR IF OK
(2) 007132 104007          ERROR 7          ;REPORT ERROR
(2) 007134 000240          705:  NOP

```

```

(1) ;*****
(1) ;CHECK THE REGISTERS STORED THROUGH PORT A. ALL REGISTERS SHOULD BE ZERO.
(1) ;THE REGISTERS ARE STORED ON THE STACK.

```

```

(1) 007136 013737 001224 001234 45:  MOV  PORTA,PTNBR ;CHANGE 'PORT NUMBER' TO THE OPPOSITE PORT
(2) 007144 010037 001122          MOV  R0,$B0ADR  ;BASE ADDRESS FOR REGISTER RPER1
(2) 007150 062737 000014 001122      ADD  #RPER1,$B0ADR ;ADDRESS OF RPER1 FOR TYF_OUT
(2) 007156 012637 001126          MOV  (SP)+,$BDDAT ;CHECK THE STORED CONTENTS OF RPER1
(2) 007162 001401          BEQ  .+4         ;CONTENTS ZERO ?
(2) 007164 104006          ERROR 6         ;REPORT THAT PORT A SAW NON-ZERO REGISTER
(2) 007166 010037 001122          MOV  R0,$B0ADR  ;BASE ADDRESS FOR REGISTER RPMR
(2) 007172 062737 000024 001122      ADD  #RPMR,$B0ADR ;ADDRESS OF RPMR FOR TYPEOUT
(2) 007200 012637 001126          MOV  (SP)+,$BDDAT ;CHECK THE STORED CONTENTS OF RPMR
(2) 007204 001401          BEQ  .+4         ;CONTENTS ZERO ?
(2) 007206 104006          ERROR 6         ;REPORT THAT PORT A SAW NON-ZERO REGISTER
(2) 007210 010037 001122          MOV  R0,$B0ADR  ;BASE ADDRESS FOR REGISTER RPDA
(2) 007214 062737 000006 001122      ADD  #RPDA,$B0ADR ;ADDRESS OF RPDA FOR TYPEOUT
(2) 007222 012637 001126          MOV  (SP)+,$BDDAT ;CHECK THE STORED CONTENTS OF RPDA
(2) 007226 001401          BEQ  .+4         ;CONTENTS ZERO ?
(2) 007230 104006          ERROR 6         ;REPORT THAT PORT A SAW NON-ZERO REGISTER
(2) 007232 010037 001122          MOV  R0,$B0ADR  ;BASE ADDRESS FOR REGISTER RPDT
(2) 007236 062737 000026 001122      ADD  #RPDT,$B0ADR ;ADDRESS OF RPDT FOR TYPEOUT
(2) 007244 012637 001126          MOV  (SP)+,$BDDAT ;CHECK THE STORED CONTENTS OF RPDT
(2) 007250 001401          BEQ  .+4         ;CONTENTS ZERO ?
(2) 007252 104006          ERROR 6         ;REPORT THAT PORT A SAW NON-ZERO REGISTER
(2) 007254 010037 001122          MOV  R0,$B0ADR  ;BASE ADDRESS FOR REGISTER RPLA
(2) 007260 062737 000020 001122      ADD  #RPLA,$B0ADR ;ADDRESS OF RPLA FOR TYPEOUT
(2) 007266 012637 001126          MOV  (SP)+,$BDDAT ;CHECK THE STORED CONTENTS OF RPLA
(2) 007272 001401          BEQ  .+4         ;CONTENTS ZERO ?
(2) 007274 104006          ERROR 6         ;REPORT THAT PORT A SAW NON-ZERO REGISTER
(2) 007276 010037 001122          MOV  R0,$B0ADR  ;BASE ADDRESS FOR REGISTER RPER2
(2) 007302 062737 000040 001122      ADD  #RPER2,$B0ADR ;ADDRESS OF RPER2 FOR TYPEOUT
(2) 007310 012637 001126          MOV  (SP)+,$BDDAT ;CHECK THE STORED CONTENTS OF RPER2
(2) 007314 001401          BEQ  .+4         ;CONTENTS ZERO ?
(2) 007316 104006          ERROR 6         ;REPORT THAT PORT A SAW NON-ZERO REGISTER
(2) 007320 010037 001122          MOV  R0,$B0ADR  ;BASE ADDRESS FOR REGISTER RPOF
(2) 007324 062737 000032 001122      ADD  #RPOF,$B0ADR ;ADDRESS OF RPOF FOR TYPEOUT
(2) 007332 012637 001126          MOV  (SP)+,$BDDAT ;CHECK THE STORED CONTENTS OF RPOF
(2) 007336 001401          BEQ  .+4         ;CONTENTS ZERO ?
(2) 007340 104006          ERROR 6         ;REPORT THAT PORT A SAW NON-ZERO REGISTER
(2) 007342 010037 001122          MOV  R0,$B0ADR  ;BASE ADDRESS FOR REGISTER RPCA
(2) 007346 062737 000034 001122      ADD  #RPCA,$B0ADR ;ADDRESS OF RPCA FOR TYPEOUT
(2) 007354 012637 001126          MOV  (SP)+,$BDDAT ;CHECK THE STORED CONTENTS OF RPCA
(2) 007360 001401          BEQ  .+4         ;CONTENTS ZERO ?
(2) 007362 104006          ERROR 6         ;REPORT THAT PORT A SEES NON-ZERO REGISTER
(2) 007364 010037 001122          MOV  R0,$B0ADR  ;BASE ADDRESS FOR REGISTER RPCC
(2) 007370 062737 000036 001122      ADD  #RPCC,$B0ADR ;ADDRESS OF RPCC FOR TYPEOUT

```

```

(2) 007376 012637 001126 MOV (SP)+, $BDDAT ;CHECK THE STORED CONTENTS OF RPEC
(2) 007402 001401 BEQ .+4 ;CONTENTS ZERO ?
(2) 007404 104006 ERROR 6 ;REPORT THAT PORT A SEES NON-ZERO REGISTER
(2) 007406 010037 001122 MOV R0, $BDADR ;BASE ADDRESS FOR REGISTER RPSN
(2) 007412 062737 000030 001122 ADD #RPSN, $BDADR ;ADDRESS OF RPSN FOR TYPEOUT
(2) 007420 012637 001126 MOV (SP)+, $BDDAT ;CHECK THE STORED CONTENTS OF RPSN
(2) 007424 001401 BEQ .+4 ;CONTENTS ZERO ?
(2) 007426 104006 ERROR 6 ;REPORT THAT PORT A SEES NON-ZERO REGISTER
(2) 007430 010037 001122 MOV R0, $BDADR ;BASE ADDRESS FOR REGISTER RPER3
(2) 007434 062737 000042 001122 ADD #RPER3, $BDADR ;ADDRESS OF RPER3 FOR TYPEOUT
(2) 007442 012637 001126 MOV (SP)+, $BDDAT ;CHECK THE STORED CONTENTS OF RPER3
(2) 007446 001401 BEQ .+4 ;CONTENTS ZERO ?
(2) 007450 104006 ERROR 6 ;REPORT THAT PORT A SEES NON-ZERO REGISTER
(2) 007452 010037 001122 MOV R0, $BDADR ;BASE ADDRESS FOR REGISTER RPEC1
(2) 007456 062737 000044 001122 ADD #RPEC1, $BDADR ;ADDRESS OF RPEC1 FOR TYPEOUT
(2) 007464 012637 001126 MOV (SP)+, $BDDAT ;CHECK THE STORED CONTENTS OF RPEC1
(2) 007470 001401 BEQ .+4 ;CONTENTS ZERO ?
(2) 007472 104006 ERROR 6 ;REPORT THAT PORT A SEES NON-ZERO REGISTER
(2) 007474 010037 001122 MOV R0, $BDADR ;BASE ADDRESS FOR REGISTER RPEC2
(2) 007500 062737 000046 001122 ADD #RPEC2, $BDADR ;ADDRESS OF RPEC2 FOR TYPEOUT
(2) 007506 012637 001126 MOV (SP)+, $BDDAT ;CHECK THE STORED CONTENTS OF RPEC2
(2) 007512 001401 BEQ .+4 ;CONTENTS ZERO ?
(2) 007514 104006 ERROR 6 ;REPORT THAT PORT A SEES NON-ZERO REGISTER
(1) 007516 000004 SS: SCOPE ;LOOP ?

```

8101
8119
8120

```

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

```

```

TST4:
TST KYBCTL ;PERFORMING ONLY SINGLE TESTS ?
BEQ 2$ ;BR IF NOT
BPL 1$ ;BR IF JUST ENTERED TEST
JMP EXEC ;RETURN & GET NEXT TEST NUMBER
1$: MOV #-1, KYBCTL ;SET SINGLE TEST INDICATOR
2$: MOV #4, $TSTNM ;TEST NUMBER
MOV #TEST4, $LPADR ;LOAD LOOP ON TEST ADDRESS
MOV #TEST4, $LPERR ;LOAD LOOP ON ERROR ADDRESS
MOV #1, $TIMES ;DO 1 ITERATION
TEST4: MOV #STACK, SP ;LOAD THE STACK POINTER
MOV #PORTA, $RPCS2(R0) ;SELECT PORT A

```

```

(2) 007520
(3) 007520 005737 001274
(3) 007524 001406
(3) 007526 100002
(3) 007530 000137 002622
(3) 007534 012737 177777 001274
(3) 007542 112737 000004 001102
(3) 007550 012737 007572 001106
(3) 007556 012737 007572 001110
(1) 007564 012737 000001 001176
8121 007572 012706 001100
8171 007576 113760 001224 000010

```

```

(2) 007604 013737 001224 001234      MOV      PORTA,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
(2)                                     ;*****
(2)                                     ;START THE TIMER
(2) 007612 005037 001252                CLR      TIME          ;CLEAR THE ELAPSED TIME COUNTER
(2) 007616 012737 003720 001254        MOV      #2000,WATCH   ;SET WATCH TO 2000 MS
(1) 007624 013737 001224 001236        MOV      PORTA,SEIZPT ;'SEIZED' PORT ADDRESS
(1)                                     ;*****
(1)                                     ;ISSUE DRIVE CLEAR COMMAND
(1) 007632 012760 000011 000000        MOV      #11,RPCS1(RO) ;ISSUE A DRIVE CLEAR
(1)                                     ;*****
(1)                                     ;VERIFY THAT DRIVE SEIZED BY PORT A.
(2) 007640 113760 001226 000010        MOV      PORTB,RPCS2(RO) ;SELECT PORT B
(2) 007646 013737 001226 001234        MOV      PORTB,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
(2) 007654 005037 001244                CLR      CKERR        ;CLEAR THE 'CHECK ERROR' INDICATOR
(2) 007660 016037 000012 001126        MOV      RPDS1(RO),SBDDAT ;GET CONTENTS OF RPDS1
(2) 007666 012737 000012 001122        MOV      #RPDS1,SBADR  ;FORM REGISTER ADDRESS OF ERROR MESSAGE
(2) 007674 060037 001122                ADD      RO,SBADR     ;ADD RH11 BASE ADDRESS
(2) 007700 005037 001124                CLR      $GDDAT       ;WHAT REGISTER SHOULD BE
(2) 007704 023737 001124 001126        CMP      $GDDAT,SBDDAT ;IS THE REGISTER OK ?
(2) 007712 001403                BEQ      64$          ;BR IF OK
(2) 007714 104012                ERROR    12           ;TYPE MESSAGE 12
(2) 007716 005137 001244                COM      CKERR        ;SET THE REGISTER COMPARE ERROR INDICATOR
(2) 007722 000240                NOP
(2) 007724 113760 001224 000010        MOV      PORTA,RPCS2(RO) ;SELECT PORT A
(2) 007732 013737 001224 001234        MOV      PORTA,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
(2) 007740 005037 001244                CLR      CKERR        ;CLEAR THE 'CHECK ERROR' INDICATOR
(2) 007744 016037 000012 001126        MOV      RPDS1(RO),SBDDAT ;GET CONTENTS OF RPDS1
(2) 007752 012737 000012 001122        MOV      #RPDS1,SBADR  ;FORM REGISTER ADDRESS OF ERROR MESSAGE
(2) 007760 060037 001122                ADD      RO,SBADR     ;ADD RH11 BASE ADDRESS
(2) 007764 012737 011600 001124        MOV      #MOL!PGM!DPR!DRY,$GDDAT ;WHAT REGISTER SHOULD BE
(2) 007772 013737 001126 001164        MOV      $BDDAT,$TMP0 ;MOVE REGISTER CONTENTS TO '$TMP0'
(2) 010000 042737 106177 001164        BIC      #71600,$TMP0 ;SAVE SPECIFIED BITS
(2) 010006 023737 001124 001164        CMP      $GDDAT,$TMP0 ;COMPARE THE BITS
(2) 010014 001414                BEQ      66$          ;BR IF OK
(2) 010016 013737 001126 001174        MOV      $BDDAT,$TMP4 ;COPY 'BAD DATA'
(2) 010024 042737 071600 001174        BIC      #71600,$TMP4 ;CLEAR THE MASKED BITS
(2) 010032 053737 001174 001124        BIS      $TMP4,$GDDAT ;'OR' WITH GOOD DATA FOR TYPEOUT
(2) 010040 104010                ERROR    10           ;REPORT THE ERROR
(2) 010042 005137 001244                COM      CKERR        ;SET THE REGISTER COMPARE ERROR INDICATOR
(2) 010046 000240                NOP
(2) 010050 005037 001244                CLR      CKERR        ;CLEAR THE 'CHECK ERROR' INDICATOR
(2) 010054 016037 000000 001126        MOV      RPDS1(RO),SBDDAT ;GET CONTENTS OF RPDS1
(2) 010062 012737 000000 001122        MOV      #RPDS1,SBADR  ;FORM REGISTER ADDRESS OF ERROR MESSAGE
(2) 010070 060037 001122                ADD      RO,SBADR     ;ADD RH11 BASE ADDRESS
(2) 010074 012737 004210 001124        MOV      #4210,$GDDAT ;WHAT REGISTER SHOULD BE
(2) 010102 013737 001126 001164        MOV      $BDDAT,$TMP0 ;MOVE REGISTER CONTENTS TO '$TMP0'
(2) 010110 042737 100000 001164        BIC      #77777,$TMP0 ;SAVE SPECIFIED BITS
(2) 010116 023737 001124 001164        CMP      $GDDAT,$TMP0 ;COMPARE THE BITS
(2) 010124 001414                BEQ      68$          ;BR IF OK
(2) 010126 013737 001126 001174        MOV      $BDDAT,$TMP4 ;COPY 'BAD DATA'

```

C05

CZRJECO, DL CTRLR LGC MACY11 30A(1052) 28-DEC-77 10:17 PAGE 66-20
 CZRJEC.P11 21-DEC-77 14:19 T4 PORT 'A' COMMAND SEIZE TEST & SET 'VV-A'

SEG 0054

```

(2) 010134 042737 077777 001174      BIC      #77777,$TMP4      ;CLEAR THE MASKED BITS
(2) 010142 053737 001174 001124      BIS      $TMP4,$GDDAT     ;'OR' WITH GOOD DATA FOR TYPEOUT
(2) 010150 104010                ERROR    10                ;REPORT THE ERROR
(2) 010152 005137 001244                COM      CKERR            ;SET THE REGISTER COMPARE ERROR INDICATOR
(2) 010156 000240                68$:    NOP

;*****
;ISSUE READIN PRESET COMMAND AND SET FMT22
(1)
(1)
(1) 010160 012760 000023 000000      MOV      #23,RPCS1(RO)    ;ISSUE A READIN PRESET
(1) 010166 012760 010000 000032      MOV      #FMT22,RPOF(RO) ;SET FMT22

;*****
;VERIFY THAT THE DRIVE STATUS IS CORRECT
(1)
(2) 010174 005037 001244                CLR      CKERR            ;CLEAR THE 'CHECK ERROR' INDICATOR
(2) 010200 016037 000012 001126      MOV      RPDS1(RO),$BDDAT ;GET CONTENTS OF RPDS1
(2) 010206 012737 000012 001122      MOV      #RPDS1,$B0ADR    ;FORM REGISTER ADDRESS OF ERROR MESSAGE
(2) 010214 060037 001122                ADD      RO,$B0ADR        ;ADD RH11 BASE ADDRESS
(2) 010220 012737 011700 001124      MOV      #MOL:PGM:DPR:DRY:VV,$GDDAT ;WHAT REGISTER SHOULD BE
(2) 010226 013737 001126 001164      MOV      $BDDAT,$TMP0     ;MOVE REGISTER CONTENTS TO 'TMP0'
(2) 010234 042737 106077 001164      BIC      #1C71700,$TMP0   ;SAVE SPECIFIED BITS
(2) 010242 023737 001124 001164      CMP      $GDDAT,$TMP0     ;COMPARE THE BITS
(2) 010250 001414                BEQ      70$              ;BR IF OK
(2) 010252 013737 001126 001174      MOV      $BDDAT,$TMP4     ;COPY 'BAD DATA'
(2) 010260 042737 071700 001174      BIC      #71700,$TMP4     ;CLEAR THE MASKED BITS
(2) 010266 053737 001174 001124      BIS      $TMP4,$GDDAT     ;'OR' WITH GOOD DATA FOR TYPEOUT
(2) 010274 104013                ERROR    13                ;TYPE MESSAGE 13
(2) 010276 005137 001244                COM      CKERR            ;SET THE REGISTER COMPARE ERROR INDICATOR
(2) 010302 000240                70$:    NOP
(2) 010304 113760 001226 000010      MOV      PORTB,RPCS2(RO)  ;SELECT PORT B
(2) 010312 013737 001226 001234      MOV      PORTB,PTNBR      ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT

;*****
;WAIT FOR TIMEOUT TO RELEASE DRIVE
(1)
(1)
(1) 010320 005760 000012                1$:    TST      RPDS1(RO)      ;WAIT FOR THE PORT TO TIME OUT
(1) 010324 001006                BNE      2$              ;BR WHEN TIMEOUT OCCURS
(1) 010326 005737 001254                TST      WATCH           ;CHECK THE WATCH
(1) 010332 001372                BNE      1$              ;BR IF NOT ZERO
(1) 010334 104036                ERROR    36                ;NO TIMEOUT WITHIN 2 SECONDS
(1) 010336 000137 010654                JMP      3$              ;BYPASS ATTN REGISTER CHECK

;*****
;SEE IF DRIVE RETURNED TO NEUTRAL
(1)
(1) 010342                2$:
(2)
(2) ;VERIFY THAT THE DRIVE IS IN NEUTRAL
(2) 010342 005037 001250                CLR      RELERR           ;CLEAR THE 'RELEASE ERROR' INDICATOR
(2) 010346 012737 000012 001122      MOV      #RPDS1,$B0ADR    ;FORM THE ADDRESS OF RPDS1 FOR TYPEOUT
(2) 010354 060037 001122                ADD      RO,$B0ADR        ;ADD THE I/O BASE ADDRESS
(2) 010360 012737 011600 001124      MOV      #MOL:PGM:DPR:DRY,$GDDAT ;COMPARISON CONSTANT
(2) 010366 113760 001224 000010      MOV      PORTA,RPCS2(RO)  ;SELECT PORT A.
(2) 010374 016037 000012 001170      MOV      RPDS1(RO),$TMP2  ;GET THE DRIVE STATUS REGISTER FROM PORT A.
    
```

D05

CZRJECO, DL CTRLR LGC MACY11 30A(1052) 28-DEC-77 10:17 PAGE 66-21
CZRJEC.P11 21-DEC-77 14:19 T4 PORT 'A' COMMAND SEIZE TEST & SET 'VV-A'

SEG 0055

```

(2) 010402 013737 001170 001164 MOV $TMP2,$TMP0 ;COPY IT INTO '$TMP0'
(2) 010410 042737 100100 001164 BIC #ATA!VV,$TMP0 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
(2) 010416 113760 001226 000010 MOV#B PORTB,RPCS2(RO) ;SELECT PORT B.
(2) 010424 016037 000012 001172 MOV RPDS1(RO),$TMP3 ;GET THE DRIVE STATUS REGISTER FROM PORT B.
(2) 010432 013737 001172 001166 MOV $TMP3,$TMP1 ;COPY IT INTO '$TMP1'
(2) 010440 042737 100100 001166 BIC #ATA!VV,$TMP1 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
(2) 010446 023737 001164 001166 CMP $TMP0,$TMP1 ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
(2) 010454 001006 BNE 72$ ;BR IF NOT
(2) 010456 005737 001164 TST $TMP0 ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
(2) 010462 001037 BNE 74$ ;BR IF NOT
(2) 010464 104046 ERROR 46 ;REPORT DRIVE NOT IN NEUTRAL OR NOT SEIZED
(2) 010466 000137 010652 JMP 76$ ;BYPASS THE REST OF THE CHECKS
(2) 010472 013737 001170 001126 72$: MOV $TMP2,$BDDAT ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
(2) 010500 013737 001226 001234 MOV PORTB,PTNBR ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
(2) 010506 113760 001226 000010 MOV#B PORTB,RPCS2(RO) ;SELECT PORT B.
(2) 010514 005737 001164 TST $TMP0 ;SEE IF STATUS EQ 0 FROM 'ORT A.
(2) 010520 001414 BEQ 73$ ;BR IF ZERO
(2) 010522 013737 001224 001234 MOV PORTA,PTNBR ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
(2) 010530 013737 001172 001126 MOV $TMP3,$BDDAT ;'BAD DATA' FOR ERROR TYPE OUT
(2) 010536 113760 001224 000010 MOV#B PORTA,RPCS2(RO) ;SELECT PORT A.
(2) 010544 005737 001166 TST $TMP1 ;SEE IF STATUS EQ ZERO FROM PORT B.
(2) 010550 001004 BNE 74$ ;BR IF NOT
(2) 010552 012737 177777 001250 73$: MOV #-1,RELERR ;SET 'RELEASE ERROR' INDICATOR
(2) 010560 104022 ERROR 22 ;TYPE ERROR MESSAGE 22
(2) 010562 013737 001170 001126 74$: MOV $TMP2,$BDDAT ;LOOK FOR BIT FAILURES WHEN RPDS1 READ
(2) 010570 013737 001224 001234 MOV PORTA,PTNBR ;CHANGE PORT NUMBER
(2) 010576 042737 100100 001170 BIC #ATA!VV,$TMP2 ;DON'T CHECK ATTN BIT OR VV BIT
(2) 010604 023737 001124 001170 CMP $GDDAT,$TMP2 ;ALL BITS OK ?
(2) 010612 001401 BEQ 75$ ;BR IF OK FROM PORT A.
(2) 010614 104007 ERROR 7 ;REPORT ERROR
(2) 010616 013737 001172 001126 75$: MOV $TMP3,$BDDAT ;CHECK RPDS1 FOR BIT FAILURES - FROM PORT B.
(2) 010624 013737 001226 001234 MOV PORTB,PTNBR ;CHANGE PORT NUMBER
(2) 010632 042737 100100 001172 BIC #ATA!VV,$TMP3 ;DON'T CHECK ATTN BIT OR VV BIT
(2) 010640 023737 001124 001172 CMP $GDDAT,$TMP3 ;SEE IF READ OK FROM PORT B.
(2) 010646 001401 BEQ 76$ ;BR IF OK
(2) 010650 104007 ERROR 7 ;REPORT ERROR
(2) 010652 000240 76$: NOP
(1) 010654 000004 3$: SCOPE ;LOOP ?

```

01000
01000

```

*****
*TEST 5 PORT 'B' COMMAND SEIZE TEST & SET 'VV-B'
*
*VERIFY THAT THE DRIVE IS SEIZED WHEN A COMMAND IS ISSUED. SET 'VV'
* FOR THE PORT UNDER TEST.
*
* A. ISSUE A DRIVE CLEAR COMMAND THROUGH PORT 'B'. VERIFY THAT THE
* DRIVE WAS SEIZED BY PORT 'B' AND THAT THE 'GO' BIT RESET.
*
* B. ISSUE A READIN PRESET COMMAND THROUGH PORT 'B'. VERIFY THAT THE
* 'VV' BIT FOR PORT 'B' WAS SET.
*
* C. STALL FOR 2 SECONDS THEN VERIFY THAT THE PORT TIMEOUT RELEASED
* THE DRIVE AND THE THE DRIVE RETURNED TO NEUTRAL.
*
*****
†ST5:

```

(2) 010656

E05

CZRJECO DL CTRLR LGC MACY11 30A(1052) 28-DEC-77 10:17 PAGE 66-22
 CZRJEC.P11 21-DEC-77 14:19 TS PORT 'B' COMMAND SEIZE TEST & SET 'VV-B'

SEQ 0056

```

(3) 010656 005737 001274          TST      KYBCTL          ;PERFORMING ONLY SINGLE TESTS ?
(3) 010662 001406                BEQ      2$             ;BR IF NOT
(3) 010664 100002                BPL      1$             ;BR IF JUST ENTERED TEST
(3) 010666 000137 002622          JMP      EXEC           ;RETURN & GET NEXT TEST NUMBER
(3) 010672 012737 177777 0C1274 1$:  MOV      #-1,KYBCTL     ;SET SINGLE TEST INDICATOR
(3) 010700 112737 000005 001102 2$:  MOVVB   #5,$TSTNM      ;TEST NUMBER
(3) 010706 012737 010730 001106    MOV      #TEST5,$LPADR  ;LOAD LOOP ON TEST ADDRESS
(3) 010714 012737 010730 001110    MOV      #TEST5,$LPERR  ;LOAD LOOP ON ERROR ADDRESS
(1) 010722 012737 000001 001176    MOV      #1,$TIMES     ;DO 1 ITERATION
8190 010730 012706 001100          TEST5: MOV      #STACK,SP ;LOAD THE STACK POINTER
8191 010734 113760 001226 000010    MOVVB   PORTB,APCS2(RO) ;SELECT PORT B
(2) 010742 013737 001226 001234    MOV      PORTB,PTNBR   ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT

;*****
;START THE TIMER

(2) 010750 005037 001252          CLR      TIME          ;CLEAR THE ELAPSED TIME COUNTER
(2) 010754 012737 003720 001254    MOV      #2000,WATCH   ;SET WATCH TO 2000 MS
(1) 010762 013737 001226 001236    MOV      PORTB,$SEIZPT ;'SEIZED' PORT ADDRESS

;*****
;ISSUE DRIVE CLEAR COMMAND

(1) 010770 012760 000011 000000    MOV      #11,APCS1(RO) ;ISSUE A DRIVE CLEAR

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

(2) 010776 113760 001224 000010    MOVVB   PORTA,APCS2(RO) ;SELECT PORT A
(2) 011004 013737 001224 001234    MOV      PORTA,PTNBR   ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
(2) 011012 005037 001244          CLR      CKERR         ;CLEAR THE 'CHECK ERROR' INDICATOR
(2) 011016 016037 000012 001126    MOV      RPDS1(RO), $BDDAT ;GET CONTENTS OF RPDS1
(2) 011024 012737 000012 001122    MOV      #RPDS1,$BDAOR  ;FORM REGISTER ADDRESS OF ERROR MESSAGE
(2) 011032 060037 001122          ADD      RO,$BDAOR     ;ADD RH11 BASE ADDRESS
(2) 011036 005037 001124          CLR      $GDDAT        ;WHAT REGISTER SHOULD BE
(2) 011042 023737 001124 001126    CMP      $GDDAT,$BDDAT ;IS THE REGISTER OK ?
(2) 011050 001403                BEQ      64$           ;BR IF OK
(2) 011052 104012                ERROR   12            ;TYPE MESSAGE 12
(2) 011054 005137 001244          COM      CKERR         ;SET THE REGISTER COMPARE ERROR INDICATOR
(2) 011060 000240                NOP

(2) 011062 113760 001226 000010    64$:  MOVVB   PORTB,APCS2(RO) ;SELECT PORT B
(2) 011070 013737 001226 001234    MOV      PORTB,PTNBR   ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
(2) 011076 005037 001244          CLR      CKERR         ;CLEAR THE 'CHECK ERROR' INDICATOR
(2) 011102 016037 000012 001126    MOV      RPDS1(RO), $BDDAT ;GET CONTENTS OF RPDS1
(2) 011110 012737 000012 001122    MOV      #RPDS1,$BDAOR  ;FORM REGISTER ADDRESS OF ERROR MESSAGE
(2) 011116 060037 001122          ADD      RO,$BDAOR     ;ADD RH11 BASE ADDRESS
(2) 011122 012737 011600 001124    MOV      #MOL!PGM!DPR!DRY,$GDDAT ;WHAT REGISTER SHOULD BE
(2) 011130 013737 001126 001164    MOV      $BDDAT,$TMP0   ;MOVE REGISTER CONTENTS TO '$TMP0'
(2) 011136 042737 106177 001164    BIC      #71600,$TMP0  ;SAVE SPECIFIED BITS
(2) 011144 023737 001124 001164    CMP      $GDDAT,$TMP0  ;COMPARE THE BITS
(2) 011152 001414                BEQ      66$           ;BR IF OK
(2) 011154 013737 001126 001174    MOV      $BDDAT,$TMP4   ;COPY 'BAD DATA'
(2) 011162 042737 071600 001174    BIC      #71600,$TMP4  ;CLEAR THE MASKED BITS
(2) 011170 053737 001174 001124    BIS      $TMP4,$GDDAT  ;'OR' WITH GOOD DATA FOR TYPEOUT
(2) 011176 104010                ERROR   10            ;REPORT THE ERROR
(2) 011200 005137 001244          COM      CKERR         ;SET THE REGISTER COMPARE ERROR INDICATOR
    
```

F05

```

(2) 011204 000240 66$: NOP
(2) 011206 005037 001244 CLR CKERR ;CLEAR THE 'CHECK ERROR' INDICATOR
(2) 011212 016037 000000 001126 MOV RPCS1(RC), $BDDAT ;GET CONTENTS OF RPCS1
(2) 011220 012737 000000 001122 MOV #RPCS1, $BDADR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
(2) 011226 060037 001122 ADD R0, $BDADR ;ADD RH11 BASE ADDRESS
(2) 011232 012737 004210 001124 MOV #4210, $GDDAT ;WHAT REGISTER SHOULD BE
(2) 011240 013737 001126 001164 MOV $BDDAT, $TMP0 ;MOVE REGISTER CONTENTS TO '$TMP0'
(2) 011246 042737 100000 001164 BIC #1C7777, $TMP0 ;SAVE SPECIFIED BITS
(2) 011254 023737 001124 001164 CMP $GDDAT, $TMP0 ;COMPARE THE BITS
(2) 011262 001414 BEQ 68$ ;BR IF OK
(2) 011264 013737 001126 001174 MOV $BDDAT, $TMP4 ;COPY 'BAD DATA'
(2) 011272 042737 077777 001174 BIC #7777, $TMP4 ;CLEAR THE MASKED BITS
(2) 011300 053737 001174 001124 BIS $TMP4, $GDDAT ;'OR' WITH GOOD DATA FOR TYPEOUT
(2) 011306 104010 ERROR 10 ;REPORT THE ERROR
(2) 011310 005137 001244 COM CKERR ;SET THE REGISTER COMPARE ERROR INDICATOR
(2) 011314 000240 68$: NOP

(1)
(2) ;*****
(1) ;ISSUE READIN PRESET COMMAND AND SET FMT22
(1)
(1) 011316 012760 000023 000000 MOV #23, RPCS1(R0) ;ISSUE A READIN PRESET
(1) 011324 012760 010000 000032 MOV #FMT22, RPOF(R0) ;SET FMT22
8195
(2) ;*****
(1) ;VERIFY THAT THE DRIVE STATUS IS CORRECT
(1)
(2) 011332 005037 001244 CLR CKERR ;CLEAR THE 'CHECK ERROR' INDICATOR
(2) 011336 016037 000012 001126 MOV RPDS1(R0), $BDDAT ;GET CONTENTS OF RPDS1
(2) 011344 012737 000012 001122 MOV #RPDS1, $BDADR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
(2) 011352 060037 001122 ADD R0, $BDADR ;ADD RH11 BASE ADDRESS
(2) 011356 012737 011700 001124 MOV #MOL!PGM!DPR!DRY!VV, $GDDAT ;WHAT REGISTER SHOULD BE
(2) 011364 013737 001126 001164 MOV $BDDAT, $TMP0 ;MOVE REGISTER CONTENTS TO '$TMP0'
(2) 011372 042737 106077 001164 BIC #1C71700, $TMP0 ;SAVE SPECIFIED BITS
(2) 011400 023737 001124 001164 CMP $GDDAT, $TMP0 ;COMPARE THE BITS
(2) 011406 001414 BEQ 70$ ;BR IF OK
(2) 011410 013737 001126 001174 MOV $BDDAT, $TMP4 ;COPY 'BAD DATA'
(2) 011416 042737 071700 001174 BIC #71700, $TMP4 ;CLEAR THE MASKED BITS
(2) 011424 053737 001174 001124 BIS $TMP4, $GDDAT ;'OR' WITH GOOD DATA FOR TYPEOUT
(2) 011432 104013 ERROR 13 ;TYPE MESSAGE 13
(2) 011434 005137 001244 COM CKERR ;SET THE REGISTER COMPARE ERROR INDICATOR
(2) 011440 000240 70$: NOP
(2) 011442 113760 001224 000010 MOVB PORTA, RPCS2(R0) ;SELECT PORT A
(2) 011450 013737 001224 001234 MOV PORTA, PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT

(1)
(2) ;*****
(1) ;WAIT FOR TIMEOUT TO RELEASE DRIVE
(1)
(1) 011456 005760 000012 1$: TST RPDS1(R0) ;WAIT FOR THE PORT TO TIME OUT
(1) 011462 001006 BNE 2$ ;BR WHEN TIMEOUT OCCURS
(1) 011464 005737 001254 TST WATCH ;CHECK THE WATCH
(1) 011470 001372 BNE 1$ ;BR IF NOT ZERO
(1) 011472 104036 ERROR 36 ;NO TIMEOUT WITHIN 2 SECONDS
(1) 011474 000137 012012 JMP 3$ ;BYPASS ATTN REGISTER CHECK

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

```

```

(1) 011500 25:
(2)
(2) ;VERIFY THAT THE DRIVE IS IN NEUTRAL
(2) 011500 005037 001250 CLR RELERR ;CLEAR THE 'RELEASE ERROR' INDICATOR
(2) 011504 012737 000012 001122 MOV #RPDS1,$BDDADR ;FORM THE ADDRESS OF RPDS1 FOR TYPEOUT
(2) 011512 060037 001122 ADD RD,$BDDADR ;ADD THE I/O BASE ADDRESS
(2) 011516 012737 011700 001124 MOV #MOL!PGM!DPR!DRY!VV,$GDDAT ;COMPARISON CONSTANT
(2) 011524 113760 001224 000010 MOVB PORTA,RPCS2(RD) ;SELECT PORT A.
(2) 011532 016037 000012 001170 MOV RPDS1(RD),$TMP2 ;GET THE DRIVE STATUS REGISTER FROM PORT A.
(2) 011540 013737 001170 001164 MOV $TMP2,$TMP0 ;COPY IT INTO 'TMP0'
(2) 011546 042737 100100 001164 BIC #ATA!VV,$TMP0 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
(2) 011554 113760 001226 000010 MOVB PORTB,RPCS2(RD) ;SELECT PORT B.
(2) 011562 016037 000012 001172 MOV RPDS1(RD),$TMP3 ;GET THE DRIVE STATUS REGISTER FROM PORT B.
(2) 011570 013737 001172 001166 MOV $TMP3,$TMP1 ;COPY IT INTO 'TMP1'
(2) 011576 042737 100100 001166 BIC #ATA!VV,$TMP1 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
(2) 011604 023737 001164 001166 CMP $TMP0,$TMP1 ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
(2) 011612 001006 BNE 72$ ;BR IF NOT
(2) 011614 005737 001164 TST $TMP0 ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
(2) 011620 001037 BNE 74$ ;BR IF NOT
(2) 011622 104046 ERROR 46 ;REPORT DRIVE NOT IN NEUTRAL OR NOT SEIZED
(2) 011624 000137 012010 JMP 76$ ;BYPASS THE REST OF THE CHECKS
(2) 011630 013737 001170 001126 72$: MOV $TMP2,$BDDAT ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
(2) 011636 013737 001226 001234 MOV PORTB,PTNBR ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
(2) 011644 113760 001226 000010 MOVB PORTB,RPCS2(RD) ;SELECT PORT B.
(2) 011652 005737 001164 TST $TMP0 ;SEE IF STATUS EQ 0 FROM PORT A.
(2) 011656 001414 BEQ 73$ ;BR IF ZERO
(2) 011660 013737 001224 001234 MOV PORTA,PTNBR ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
(2) 011666 013737 001172 001126 MOV $TMP3,$BDDAT ;'BAD DATA' FOR ERROR TYPE OUT
(2) 011674 113760 001224 000010 MOVB PORTA,RPCS2(RD) ;SELECT PORT A.
(2) 011702 005737 001166 TST $TMP1 ;SEE IF STATUS EQ ZERO FROM PORT B.
(2) 011706 001004 BNE 74$ ;BR IF NOT
(2) 011710 012737 177777 001250 73$: MOV #-1,RELERR ;SET 'RELEASE ERROR' INDICATOR
(2) 011716 104026 ERROR 26 ;TYPE ERROR MESSAGE 26
(2) 011720 013737 001170 001126 74$: MOV $TMP2,$BDDAT ;LOOK FOR BIT FAILURES WHEN RPDS1 READ
(2) 011726 013737 001224 001234 MOV PORTA,PTNBR ;CHANGE PORT NUMBER
(2) 011734 042737 100000 001170 BIC #ATA,$TMP2 ;DON'T CHECK THE ATTN BIT
(2) 011742 023737 001124 001170 CMP $GDDAT,$TMP2 ;ALL BITS OK ?
(2) 011750 001401 BEQ 75$ ;BR IF OK FROM PORT A.
(2) 011752 104007 ERROR 7 ;REPORT ERROR
(2) 011754 013737 001172 001126 75$: MOV $TMP3,$BDDAT ;CHECK RPDS1 FOR BIT FAILURES - FROM PORT B.
(2) 011762 013737 001226 001234 MOV PORTB,PTNBR ;CHANGE PORT NUMBER
(2) 011770 042737 100000 001172 BIC #ATA,$TMP3 ;DON'T CHECK THE ATTN BIT
(2) 011776 023737 001124 001172 CMP $GDDAT,$TMP3 ;SEE IF READ OK FROM PORT B.
(2) 012004 001401 BEQ 76$ ;BR IF OK
(2) 012006 104007 ERROR 7 ;REPORT ERROR
(2) 012010 000240 76$: NOP
(1) 012012 000004 3$: SCOPE ;LOOP ?

```

8196
8208
8209
(3)
(4)
1

```

*****
*TEST 6 TEST RELEASE, DRIVE SEIZED BY PORT 'A'
*
*TEST THE OPERATION OF THE RELEASE COMMAND, DRIVE SEIZED
*

```


(3)	012260	023737	001164	001166		CMP	\$TMP0,\$TMP1	: IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
(3)	012266	001006				BNE	66\$: BR IF NOT
(3)	012270	005737	001164			TST	\$TMP0	: REGISTERS ARE THE SAME: ARE THEY ZERO ?
(3)	012274	001037				BNE	68\$: BR IF NOT
(3)	012276	104046				ERROR	46	: REPORT DRIVE NOT IN NEUTRAL OR NOT SEIZED
(3)	012300	000137	012464			JMP	70\$: BYPASS THE REST OF THE CHECKS
(3)	012304	013737	001170	001126	66\$:	MOV	\$TMP2,\$BDDAT	: SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
(3)	012312	013737	001226	001234		MOV	PORTB,PTNBR	: SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
(3)	012320	113760	001226	000010		MOV	PORTB,RPCS2(RO)	: SELECT PORT B.
(3)	012326	005737	001164			TST	\$TMP0	: SEE IF STATUS EQ 0 FROM PORT A.
(3)	012332	001414				BEQ	67\$: BR IF ZERO
(3)	012334	013737	001224	001234		MOV	PORTA,PTNBR	: SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
(3)	012342	013737	001172	001126		MOV	\$TMP3,\$BDDAT	: 'BAD DATA' FOR ERROR TYPE OUT
(3)	012350	113760	001224	000010		MOV	PORTA,RPCS2(RO)	: SELECT PORT A.
(3)	012356	005737	001166			TST	\$TMP1	: SEE IF STATUS EQ ZERO FROM PORT B.
(3)	012362	001004				BNE	68\$: BR IF NOT
(3)	012364	012737	177777	001250	67\$:	MOV	#-1,RELERR	: SET 'RELEASE ERROR' INDICATOR
(3)	012372	104026				ERROR	26	: TYPE ERROR MESSAGE 26
(3)	012374	013737	001170	001126	68\$:	MOV	\$TMP2,\$BDDAT	: LOOK FOR BIT FAILURES WHEN RPDS1 READ
(3)	012402	013737	001224	001234		MOV	PORTA,PTNBR	: CHANGE PORT NUMBER
(3)	012410	042737	100000	001170		BIC	#ATA,\$TMP2	: DON'T CHECK THE ATTN BIT
(3)	012416	023737	001124	001170		CMP	\$GDDAT,\$TMP2	: ALL BITS OK ?
(3)	012424	001401				BEQ	69\$: BR IF OK FROM PORT A.
(3)	012426	104007				ERROR	7	: REPORT ERROR
(3)	012430	013737	001172	001126	69\$:	MOV	\$TMP3,\$BDDAT	: CHECK RPDS1 FOR BIT FAILURES - FROM PORT B.
(3)	012436	013737	001226	001234		MOV	PORTB,PTNBR	: CHANGE PORT NUMBER
(3)	012444	042737	100000	001172		BIC	#ATA,\$TMP3	: DON'T CHECK THE ATTN BIT
(3)	012452	023737	001124	001172		CMP	\$GDDAT,\$TMP3	: SEE IF READ OK FROM PORT B.
(3)	012460	001401				BEQ	70\$: BR IF OK
(3)	012462	104007				ERROR	7	: REPORT ERROR
(3)	012464	000240			70\$:	NOP		
(1)	012466	005737	001250			TST	RELERR	: DID DRIVE RETURN TO NEUTRAL ?
(1)	012472	001402				BEQ	+6	: BR IF IN NEUTRAL
(1)	012474	000137	012750			JMP	1\$: GO WAIT FOR DRIVE TO TIMEOUT
(2)	012500	113760	001224	000010		MOV	PORTA,RPCS2(RO)	: SELECT PORT A
(2)	012506	013737	001224	001234		MOV	PORTA,PTNBR	: MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
(2)	012514	005037	001244			CLR	CKERR	: CLEAR THE 'CHECK ERROR' INDICATOR
(2)	012520	016037	000012	001126		MOV	RPDS1(RO),\$BDDAT	: GET CONTENTS OF RPDS1
(2)	012526	012737	000012	001122		MOV	#RPDS1,\$BDDADR	: FORM REGISTER ADDRESS OF ERROR MESSAGE
(2)	012534	060037	001122			ADD	RO,\$BDDADR	: ADD RHL1 BASE ADDRESS
(2)	012540	005037	001124			CLR	\$GDDAT	: WHAT REGISTER SHOULD BE
(2)	012544	013737	001126	001164		MOV	\$BDDAT,\$TMP0	: MOVE REGISTER CONTENTS TO '\$TMP0'
(2)	012552	042737	077777	001164		BIC	#ICATA,\$TMP0	: SAVE SPECIFIED BITS
(2)	012560	023737	001124	001164		CMP	\$GDDAT,\$TMP0	: COMPARE THE BITS
(2)	012566	001414				BEQ	71\$: BR IF OK
(2)	012570	013737	001126	001174		MOV	\$BDDAT,\$TMP4	: COPY 'BAD DATA'
(2)	012576	042737	100000	001174		BIC	#ATA,\$TMP4	: CLEAR THE MASKED BITS
(2)	012604	053737	001174	001124		BIS	\$TMP4,\$GDDAT	: 'OR' WITH GOOD DATA FOR TYPEOUT
(2)	012612	104017				ERROR	17	: TYPE MESSAGE 17
(2)	012614	005137	001244			COM	CKERR	: SET THE REGISTER COMPARE ERROR INDICATOR
(2)	012620	000240			71\$:	NOP		
(2)	012622	113760	001226	000010		MOV	PORTB,RPCS2(RO)	: SELECT PORT B
(2)	012630	013737	001226	001234		MOV	PORTB,PTNBR	: MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
(2)	012636	005037	001244			CLR	CKERR	: CLEAR THE 'CHECK ERROR' INDICATOR
(2)	012642	016037	000012	001126		MOV	RPDS1(RO),\$BDDAT	: GET CONTENTS OF RPDS1
(2)	012650	012737	000012	001122		MOV	#RPDS1,\$BDDADR	: FORM REGISTER ADDRESS OF ERROR MESSAGE

J05

CZRJECD, DL CTRLR LGC MACY.1 30A 1052
CZRJEC.P11 21-DEC-77 14:19

28-DEC-77 10:17 PAGE 66-27
*6 TEST RELEASE, DRIVE SEIZED BY PORT 'A'

SEQ 0061

```

(2) 012656 060037 001122 ADD R0,SBDADR ;ADD RH11 BASE ADDRESS
(2) 012662 005037 001124 CLR SGDDAT ;WHAT REGISTER SHOULD BE
(2) 012666 013737 001126 001164 MOV SBDADR,$TMP0 ;MOVE REGISTER CONTENTS TO '$TMP0'
(2) 012674 042737 077777 001164 BIC #1CATA,$TMP0 ;SAVE SPECIFIED BITS
(2) 012702 027737 001124 001164 CMP SGDDAT,$TMP0 ;COMPARE THE BITS
(2) 012710 00414 BEQ 73$ ;BR IF OK
(2) 012712 013737 001126 001174 MOV SBDADR,$TMP4 ;COPY 'BAD DATA'
(2) 012720 042737 100000 001174 BIC #ATA,$TMP4 ;CLEAR THE MASKED BITS
(2) 012726 053737 001174 001124 BIS $TMP4,$GDDAT ;'OR' WITH GOOD DATA FOR TYPEOUT
(2) 012734 104017 ERROR 17 ;TYPE MESSAGE 17
(2) 012736 005137 001244 COM CKERR ;SET THE REGISTER COMPARE ERROR INDICATOR
(2) 012742 000240 73$: NOP
(1) 012744 000137 013002 JMP 2$ ;GO CHECK FOR LOOP ON ERROR

```

```

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

```

```

(1) 012750 113760 001226 000010 1$: MOVB PORTB,RPCS2(R0) ;SELECT PORT B
(2) 012750 013737 001226 001234 MOV PORTB,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
(1) 012764 005760 000012 TST RPS1(R0) ;WAIT FOR TIMEOUT TO RELEASE DRIVE
(1) 012770 001004 BNE 2$ ;BR WHEN DRIVE RELEASED
(1) 012772 005737 001254 TST WATCH ;CHECK THE WATCH
(1) 012776 001364 BNE 1$ ;BR IF NOT ZERO
(1) 013000 104036 ERROR 36 ;NO TIMEOUT WITHIN 2 SECONDS
(1) 013002 000004 2$: SCOPE ;LOOP ?

```

```

8252
8253
(3) *****
(4) *TEST 7 TEST RELEASE, DRIVE SEIZED BY PORT 'B'
(4) *
(4) *TEST THE OPERATION OF THE RELEASE COMMAND, DRIVE SEIZED
(4) *
(4) * A. SEIZE THE DRIVE THROUGH PORT 'B' BY WRITING 0'S INTO RPS1.
(4) *
(4) * B. ISSUE A RELEASE COMMAND THROUGH PORT 'B'. VERIFY THAT THE DRIVE
(4) * RETURNED TO NEUTRAL, AND THAT NO ERRORS ARE INDICATED BY THE
(4) * DRIVE.
(4) *
(3) *****

```

```

(2) 013004
(3) 013004 005737 001274 *ST7: TST KYBCTL ;PERFORMING ONLY SINGLE TESTS ?
(3) 013010 001406 BEQ 2$ ;BR IF NOT
(3) 013012 100002 BPL 1$ ;BR IF JUST ENTERED TEST
(3) 013014 000137 002622 JMP EXEC ;RETURN & GET NEXT TEST NUMBER
(3) 013020 012737 177777 001274 1$: MOV #-1,KYBCTL ;SET SINGLE TEST INDICATOR
(3) 013026 112737 000007 001102 2$: MOVB #7,$STNM ;TEST NUMBER
(3) 013034 012737 013056 001106 MOV #TEST7,$LPADR ;LOAD LOOP ON TEST ADDRESS
(3) 013042 012737 013056 001110 MOV #TEST7,$LPERR ;LOAD LOOP ON ERROR ADDRESS
(1) 013050 012737 007640 001176 MOV #4000,$TIMES ;DO 4000 ITERATIONS
8254 013056 012706 001100 TEST7: MOV #STACK,$P ;LOAD THE STACK POINTER
8255

```

```

*****
;START THE TIMER

```

```

(2) 013062 005037 001252 CLR TIME ;CLEAR THE ELAPSED TIME COUNTER

```

K05

CZRJECO, DL CTRLR LGC MACY11 30A(1052) 28-DEC-77 10:17 PAGE 66-28
 CZRJEC.P11 21-DEC-77 14:19

TEST RELEASE, DRIVE SEIZED BY PORT 'B'

SEQ 0062

```
(2) 013066 012737 003720 001254      MOV      #2000.,WATCH      ;SET WATCH TO 2000 MS
(1)
(2)
(1)
(2)
(2) 013074 113760 001226 000010      MOV      PORTB,RPCS2(RO)  ;SELECT PORT B
(2) 013102 013737 001226 001236      MOV      PORTB,SEIZPT    ;STORE SEIZING PORT'S ADDRESS
(2) 013110 005060 000012 000012      CLR      RPDS1(RO)      ;WRITE RPDS1
(2) 013114 013737 001224 001240      MOV      PORTA,OPPRT    ;'OPPOSITE' PORT ADDRESS
(1)
(2)
(1)
(2)
(2) 013122 113760 001226 000010      MOV      PORTB,RPCS2(RO)  ;SELECT PORT B
(3) 013130 013737 001226 001234      MOV      PORTB,PTNBR    ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
(2) 013136 012760 000013 000000      MOV      #13,RPDS1(RO)  ;ISSUE RELEASE THROUGH PORT B
(3)
(3)
(3)
(3)
(3) 013144 005037 001250 001122      CLR      RELERR          ;CLEAR THE 'RELEASE ERROR' INDICATOR
(3) 013150 012737 000012 001122      MOV      #RPDS1,$BDDADR  ;FORM THE ADDRESS OF RPDS1 FOR TYPEOUT
(3) 013156 060037 001122 001122      ADD     RO,$BDDADR      ;ADD THE I/O BASE ADDRESS
(3) 013162 012737 011700 001124      MOV      #MOL!PGM!DPR!DRY!VV,$GDAT ;COMPARISON CONSTANT
(3) 013170 113760 001224 000010      MOV      PORTA,RPCS2(RO) ;SELECT PORT A.
(3) 013176 016037 000012 001170      MOV      RPDS1(RO),$TMP2 ;GET THE DRIVE STATUS REGISTER FROM PORT A.
(3) 013204 013737 001170 001164      MOV      $TMP2,$TMP0    ;COPY IT INTO '$TMP0'
(3) 013212 042737 100100 001164      BIC     #ATA!VV,$TMP0   ;CLEAR PORT DEPENDENT BITS FROM THE COPY
(3) 013220 113760 001226 000010      MOV      PORTB,RPCS2(RO) ;SELECT PORT B.
(3) 013226 016037 000012 001172      MOV      RPDS1(RO),$TMP3 ;GET THE DRIVE STATUS REGISTER FROM PORT B.
(3) 013234 013737 001172 001166      MOV      $TMP3,$TMP1    ;COPY IT INTO '$TMP1'
(3) 013242 042737 100100 001166      BIC     #ATA!VV,$TMP1   ;CLEAR PORT DEPENDENT BITS FROM THE COPY
(3) 013250 023737 001164 001166      CMP     $TMP0,$TMP1     ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
(3) 013256 001006 001164 001166      BNE     66$             ;BR IF NOT
(3) 013260 005737 001164 001166      TST    $TMP0           ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
(3) 013264 001037 001164 001166      BNE     68$             ;BR IF NOT
(3) 013266 104046 001164 001166      ERROR  46              ;REPORT DRIVE NOT IN NEUTRAL OR NOT SEIZED
(3) 013270 000137 013454 001126      JMP     70$             ;BYPASS THE REST OF THE CHECKS
(3) 013274 013737 001170 001126 66$:  MOV     $TMP2,$BDDAT    ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
(3) 013302 013737 001226 001234      MOV     PORTB,PTNBR    ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
(3) 013310 113760 001226 000010      MOV     PORTB,RPCS2(RO); SELECT PORT B.
(3) 013316 005737 001164 000010      TST    $TMP0          ;SEE IF STATUS EQ 0 FROM PORT A.
(3) 013322 001414 001164 000010      BEQ    67$             ;BR IF ZERO
(3) 013324 013737 001224 001234      MOV     PORTA,PTNBR    ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
(3) 013332 013737 001172 001126      MOV     $TMP3,$BDDAT  ;'BAD DATA' FOR ERROR TYPE OUT
(3) 013340 113760 001224 000010      MOV     PORTA,RPCS2(RO); SELECT PORT A.
(3) 013346 005737 001166 000010      TST    $TMP1          ;SEE IF STATUS EQ ZERO FROM PORT B.
(3) 013352 001004 001166 000010      BNE     68$             ;BR IF NOT
(3) 013354 012737 177777 001250 67$:  MOV     #-1,RELERR     ;SET 'RELEASE ERROR' INDICATOR
(3) 013362 104026 001250 001250      ERROR  26              ;TYPE ERROR MESSAGE 26
(3) 013364 013737 001170 001126 68$:  MOV     $TMP2,$BDDAT    ;LOOK FOR BIT FAILURES WHEN RPDS1 READ
(3) 013372 013737 001224 001234      MOV     PORTA,PTNBR    ;CHANGE PORT NUMBER
(3) 013400 042737 100000 001170      BIC     #ATA,$TMP2     ;DON'T CHECK THE ATTN BIT
(3) 013406 023737 001124 001170      CMP     $GDAT,$TMP2    ;ALL BITS OK ?
(3) 013414 001401 001124 001170      BEQ    69$             ;BR IF OK FROM PORT A.
```

L05

CZRJECO, DL CTRLR LGC MACY11 30A(1052) 28-DEC-77 10:17 PAGE 66-29
CZRJEC.P11 21-DEC-77 14:19 17 TEST RELEASE, DRIVE SEIZED BY PORT 'B'

SEQ 0063

```

(3) 013416 104007          ERROR          7          ;REPORT ERROR
(3) 013420 013737 001172 001126 69$: MOV          $TMP3,$BDDAT ;CHECK RPDS1 FOR BIT FAILURES - FROM PORT B.
(3) 013426 013737 001226 001234   MOV          PORTB,PTNBR ;CHANGE PORT NUMBER
(3) 013434 042737 100000 001172   BIC          #ATA,$TMP3 ;DON'T CHECK THE ATTN BIT
(3) 013442 023737 001124 001172   CMP          $GDDAT,$TMP3 ;SEE IF READ OK FROM PORT B.
(3) 013450 001401          BEQ          70$      ;BR IF OK
(3) 013452 104007          ERROR          7          ;REPORT ERROR
(3) 013454 000240          NOP
(1) 013456 005737 001250   TST          RELERR   ;DID DRIVE RETURN TO NEUTRAL ?
(1) 013462 001402          BEQ          .+6      ;BR IF IN NEUTRAL
(1) 013464 000137 013740   JMP          1$       ;GO WAIT FOR DRIVE TO TIMEOUT
(2) 013470 113760 001226 000010   MOVB        PORTA,RPCS2(RO) ;SELECT PORT A
(2) 013476 013737 001226 001234   MOV          PORTB,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
(2) 013504 005037 001244          CLR          CKERR    ;CLEAR THE 'CHECK ERROR' INDICATOR
(2) 013510 016037 000012 001126   MOV          RPDS1(RO),$BDDAT ;GET CONTENTS OF RPDS1
(2) 013516 012737 000012 001122   MOV          #RPDS1,$BDAOR ;FORM REGISTER ADDRESS OF _RFOR MESSAGE
(2) 013524 060037 001122          ADD          RO,$BDAOR ;ADD RH11 BASE ADDRESS
(2) 013530 005037 001124          CLR          $GDDAT   ;WHAT REGISTER SHOULD BE
(2) 013534 013737 001126 001164   MOV          $BDDAT,$TMP0 ;MOVE REGISTER CONTENTS TO '$TMP0'
(2) 013542 042737 077777 001164   BIC          #ICATA,$TMP0 ;SAVE SPECIFIED BITS
(2) 013550 023737 001124 001164   CMP          $GDDAT,$TMP0 ;COMPARE THE BITS
(2) 013556 001414          BEQ          71$     ;BR IF OK
(2) 013560 013737 001126 001174   MOV          $BDDAT,$TMP4 ;COPY 'BAD DATA'
(2) 013566 042737 100000 001174   BIC          #ATA,$TMP4 ;CLEAR THE MASKED BITS
(2) 013574 053737 001174 001124   BIS          $TMP4,$GDDAT ;'OR' WITH GOOD DATA FOR TYPEOUT
(2) 013602 104017          ERROR          17     ;TYPE MESSAGE 17
(2) 013604 005137 001244          COM          CKERR   ;SET THE REGISTER COMPARE ERROR INDICATOR
(2) 013610 000240          NOP
(2) 013612 113760 001224 000010   MOVB        PORTA,RPCS2(RO) ;SELECT PORT A
(2) 013620 013737 001224 001234   MOV          PORTA,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
(2) 013626 005037 001244          CLR          CKERR   ;CLEAR THE 'CHECK ERROR' INDICATOR
(2) 013632 016037 000012 001126   MOV          RPDS1(RO),$BDDAT ;GET CONTENTS OF RPDS1
(2) 013640 012737 000012 001122   MOV          #RPDS1,$BDAOR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
(2) 013646 060037 001122          ADD          RO,$BDAOR ;ADD RH11 BASE ADDRESS
(2) 013652 005037 001124          CLR          $GDDAT   ;WHAT REGISTER SHOULD BE
(2) 013656 013737 001126 001164   MOV          $BDDAT,$TMP0 ;MOVE REGISTER CONTENTS TO '$TMP0'
(2) 013664 042737 077777 001164   BIC          #ICATA,$TMP0 ;SAVE SPECIFIED BITS
(2) 013672 023737 001124 001164   CMP          $GDDAT,$TMP0 ;COMPARE THE BITS
(2) 013700 001414          BEQ          73$     ;BR IF OK
(2) 013702 013737 001126 001174   MOV          $BDDAT,$TMP4 ;COPY 'BAD DATA'
(2) 013710 042737 100000 001174   BIC          #ATA,$TMP4 ;CLEAR THE MASKED BITS
(2) 013716 053737 001174 001124   BIS          $TMP4,$GDDAT ;'OR' WITH GOOD DATA FOR TYPEOUT
(2) 013724 104017          ERROR          17     ;TYPE MESSAGE 17
(2) 013726 005137 001244          COM          CKERR   ;SET THE REGISTER COMPARE ERROR INDICATOR
(2) 013732 000240          NOP
(1) 013734 000137 013772   JMP          2$       ;GO CHECK FOR LOOP ON ERROR
(1)
(1) ;*****
(1) ;IF RELEASE COMMAND DIDN'T RELEASE THE DRIVE, WAIT FOR THE PORT TIMEOUT
(1) ;TO RELEASE THE DRIVE
(1)
(1) 013740          1$:
(2) 013740 113760 001224 000010   MOVB        PORTA,RPCS2(RO) ;SELECT PORT A
(2) 013746 013737 001224 001234   MOV          PORTA,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
(1) 013754 005760 000012          TST          RPDS1(RO) ;WAIT FOR TIMEOUT TO RELEASE DRIVE
(1) 013760 001004          BNE          2$       ;BR WHEN DRIVE RELEASED

```


MOS

CZRJECO, DL CTRLR LGC MACY11 30A(1052) 28-DEC-77 10:17 PAGE 66-30
CZRJEC.P11 21-DEC-77 14:19 T7 TEST RELEASE, DRIVE SEIZED BY PORT 'B'

SEQ 0064

(1) 013762 005737 001254
(1) 013766 001364
(1) 013770 104036
(1) 013772 000004

TST WATCH ;CHECK THE WATCH
BNE 15 ;BR IF NOT ZERO
ERROR 36 ;NO TIMEOUT WITHIN 2 SECONDS
2S: SCOPE ;LOOP ?

8256
8265
8266

*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.
*

(2) 013774
(3) 013774 005737 001274
(3) 014000 001406
(3) 014002 100002
(3) 014004 000137 002622
(3) 014010 012737 177777 001274
(3) 014016 112737 000010 001102
(3) 014024 012737 014046 001106
(3) 014032 012737 014046 001110
(1) 014040 012737 000144 001176
8267 014046 012706 001100
8283 014052 113760 001224 000010
(2) 014060 013737 001224 001234
(1) 014066 013737 001224 001236

↑ST10:
TST KYBCTL ;PERFORMING ONLY SINGLE TESTS ?
BEQ 2S ;BR IF NOT
BPL 15 ;BR IF JUST ENTERED TEST
JMP EXEC ;RETURN & GET NEXT TEST NUMBER
1S: MOV #-1,KYBCTL ;SET SINGLE TEST INDICATOR
2S: MOVB #10,\$STNM ;TEST NUMBER
MOV #TEST10,\$LPADR ;LOAD LOOP ON TEST ADDRESS
MOV #TEST10,\$LPERR ;LOAD LOOP ON ERROR ADDRESS
MOV #100,\$TIMES ;DO 100. ITERATIONS
TEST10: MOV #STACK,\$SP ;LOAD THE STACK POINTER
MOVB PORTA,\$PC52(RO) ;SELECT PORT A
MOV PORTA,\$PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
MOV PORTA,\$SEIZPT ;ADDR OF PORT WHICH WILL ISSUE RELEASE

(1)
(2)
(1)
(1) 014074 012760 000013 000000

;ISSUE A RELEASE COMMAND
MOV #13,\$PC51(RO) ;ISSUE A RELEASE COMMAND

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

;VERIFY THAT THE DRIVE IS STILL IN NEUTRAL

;VERIFY THAT THE DRIVE IS IN NEUTRAL

(2) 014102 005037 001250
(2) 014106 012737 000012 001122
(2) 014114 060037 001122
(2) 014120 012737 011700 001124
(2) 014126 113760 001224 000010
(2) 014134 016037 000012 001170
(2) 014142 013737 001170 001164
(2) 014150 042737 100100 001164
(2) 014156 113760 001226 000010
(2) 014164 016037 000012 001172
(2) 014172 013737 001172 001166
(2) 014200 042737 100100 001166
(2) 014206 023737 001164 001166
(2) 014214 001006
(2) 014216 005737 001164

CLR RELERR ;CLEAR THE 'RELEASE ERROR' INDICATOR
MOV #RPDS1,\$BDAADR ;FORM THE ADDRESS OF RPDS1 FOR TYPEOUT
ADD RO,\$BDAADR ;ADD THE I/O BASE ADDRESS
MOV #MOL,\$PGM;\$DPR;\$DRY;\$VV,\$GDDAT ;COMPARISON CONSTANT
MOVB PORTA,\$PC52(RO) ;SELECT PORT A.
MOV RPDS1(RO),\$STMP2 ;GET THE DRIVE STATUS REGISTER FROM PORT A.
MOV \$STMP2,\$STMP0 ;COPY IT INTO 'STMP0'
BIC #ATA!VV,\$STMP0 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
MOVB PORTB,\$PC52(RO) ;SELECT PORT B.
MOV RPDS1(RO),\$STMP3 ;GET THE DRIVE STATUS REGISTER FROM PORT B.
MOV \$STMP3,\$STMP1 ;COPY IT INTO 'STMP1'
BIC #ATA!VV,\$STMP1 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
CMP \$STMP0,\$STMP1 ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
BNE 64S ;BR IF NOT
TST \$STMP0 ;REGISTERS ARE THE SAME: ARE THEY ZERO ?

N05

CZRJECO DL CTRLR LGC MACY11 30A(1052) 28-DEC-77 10:17 PAGE 66-31
CZRJEC P11 21-DEC-77 14:19 T10 TEST RELEASE THROUGH PORT 'A', DRIVE IN NEUTRAL

SEQ 0065

```

(2) 014222 001037 BNE 66$ ;BR IF NOT
(2) 014224 104046 ERROR 46 ;REPORT DRIVE NOT IN NEUTRAL OR NOT SEIZED
(2) 014226 000137 014412 JMP 68$ ;BYPASS THE REST OF THE CHECKS
(2) 014232 013737 001170 001126 64$: MOV $TMP2,$BDDAT ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
(2) 014240 013737 001226 001234 MOV PORTB,PTNBR ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
(2) 014246 113760 001226 000010 MOVB PORTB,RPCS2(RO) ;SELECT PORT B.
(2) 014254 005737 001164 TST $TMP0 ;SEE IF STATUS EQ 0 FROM PORT A.
(2) 014260 001414 BEQ 65$ ;BR IF ZERO
(2) 014262 013737 001224 001234 MOV PORTA,PTNBR ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
(2) 014270 013737 001172 001126 MOV $TMP3,$BDDAT ;'BAD DATA' FOR ERROR TYPE OUT
(2) 014276 113760 001224 000010 MOV PORTA,RPCS2(RO) ;SELECT PORT A.
(2) 014304 005737 001166 TST $TMP1 ;SEE IF STATUS EQ ZERO FROM PORT B.
(2) 014310 001004 BNE 66$ ;BR IF NOT
(2) 014312 012737 177777 001250 65$: MOV #-1,RELEA ;SET 'RELEASE ERROR' INDICATOR
(2) 014320 104030 ERROR 30 ;TYPE ERROR MESSAGE 30
(2) 014322 013737 001170 001126 66$: MOV $TMP2,$BDDAT ;LOOK FOR BIT FAILURES WITHIN RPS1 READ
(2) 014330 013737 001224 001234 MOV PORTA,PTNBR ;CHANGE PORT NUMBER
(2) 014336 042737 100000 001170 BIC #ATA,$TMP2 ;DON'T CHECK THE ATTN BIT
(2) 014344 023737 001124 001170 CMP $GDDAT,$TMP2 ;ALL BITS OK ?
(2) 014352 001401 BEQ 67$ ;BR IF OK FROM PORT A.
(2) 014354 104007 ERROR 7 ;REPORT ERROR
(2) 014356 013737 001172 001126 67$: MOV $TMP3,$BDDAT ;CHECK RPS1 FOR BIT FAILURES - FROM PORT B.
(2) 014364 013737 001226 001234 MOV PORTB,PTNBR ;CHANGE PORT NUMBER
(2) 014372 042737 100000 001172 BIC #ATA,$TMP3 ;DON'T CHECK THE ATTN BIT
(2) 014400 023737 001124 001172 CMP $GDDAT,$TMP3 ;SEE IF READ OK FROM PORT B.
(2) 014406 001401 BEQ 68$ ;BR IF OK
(2) 014410 104007 ERROR 7 ;REPORT ERROR
(2) 014412 000240 68$: NOP
(1) 014414 000004 SCOPE ;LOOP ?

```

```

8292
8293
(3) *****
(4) *TEST 11 TEST RELEASE THROUGH PORT 'B', DRIVE IN NEUTRAL
(4) *
(4) *TEST OPERATION OF RELEASE COMMAND, DRIVE IN NEUTRAL
(4) *
(4) * A. ISSUE A RELEASE COMMAND THROUGH PORT 'B' WITH THE DRIVE IN
(4) * NEUTRAL; VERIFY THAT THE DRIVE REMAINS IN NEUTRAL.
(4) *
(3) *****
(2) ST11:
(3) 014416 005737 001274 TST KYBCTL ;PERFORMING ONLY SINGLE TESTS ?
(3) 014422 001406 BEQ 2$ ;BR IF NOT
(3) 014424 100002 BPL 1$ ;BR IF JUST ENTERED TEST
(3) 014426 000137 002622 JMP EXEC ;RETURN & GET NEXT TEST NUMBER
(3) 014432 012737 177777 001274 1$: MOV #-1,KYBCTL ;SET SINGLE TEST INDICATOR
(3) 014440 112737 000011 001102 2$: MOVB #11,$STNM ;TEST NUMBER
(3) 014446 012737 014470 001106 MOV $TEST11,$LPADR ;LOAD LOOP ON TEST ADDRESS
(3) 014454 012737 014470 001110 MOV $TEST11,$LPERR ;LOAD LOOP ON ERROR ADDRESS
(1) 014462 012737 000144 001176 MOV #100,$TIMES ;DO 100 ITERATIONS
8294 014470 012706 001100 TEST11: MOV $STACK,SP ;LOAD THE STACK POINTER
8295 014474 113760 001226 000010 MOVB PORTB,RPCS2(RO) ;SELECT PORT B
(2) 014502 013737 001226 001234 MOV PORTB,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
(1) 014510 013737 001226 001236 MOV PORTB,SEIZPT ;ADDR OF PORT WHICH WILL ISSUE RELEASE
(1) *****
(2) :ISSUE A RELEASE COMMAND
(1)

```

```

(1) (1) 014516 012760 000013 000000      MOV      #13,RPCS1(RO) ;ISSUE A RELEASE COMMAND
(1) (1)                                     ;*****
(2) (2)                                     ;VERIFY THAT THE DRIVE IS STILL IN NEUTRAL
(1) (1)                                     ;
(2) (2)                                     ;VERIFY THAT THE DRIVE IS IN NEUTRAL
(2) (2) 014524 005037 001250              CLR      RELERR        ;CLEAR THE 'RELEASE ERROR' INDICATOR
(2) (2) 014530 012737 000012 001122      MOV      #RPOS1,$B0ADR ;FORM THE ADDRESS OF RPDS1 FOR TYPEOUT
(2) (2) 014536 060037 001122              ADD      RO,$B0ADR     ;ADD THE I/O BASE ADDRESS
(2) (2) 014542 012737 011700 001124      MOV      #MOL!PGM!DPR!DRY!VV,$GDDAT ;COMPARISON CONSTANT
(2) (2) 014550 113760 001224 000010      MOV      PORTA,RPCS2(RO) ;SELECT PORT A.
(2) (2) 014556 016037 000012 001170      MOV      RPDS1(RO),STMP2 ;GET THE DRIVE STATUS REGISTER FROM PORT A.
(2) (2) 014564 013737 001170 001164      MOV      STMP2,STMP0   ;COPY IT INTO 'STMP0'
(2) (2) 014572 042737 100100 001164      BIC      #ATA!VV,STMP0 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
(2) (2) 014600 113760 001226 000010      MOV      PORTB,RPCS2(RO) ;SELECT PORT B.
(2) (2) 014606 016037 000012 001172      MOV      RPDS1(RO),STMP3 ;GET THE DRIVE STATUS REGISTER FROM PORT B.
(2) (2) 014614 013737 001172 001166      MOV      STMP3,STMP1   ;COPY IT INTO 'STMP1'
(2) (2) 014622 042737 100100 001166      BIC      #ATA!VV,STMP1 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
(2) (2) 014630 023737 001164 001166      CMP      STMP0,STMP1   ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
(2) (2) 014636 001006                      BNE      64$           ;BR IF NOT
(2) (2) 014640 005737 001164                      TST     STMP0         ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
(2) (2) 014644 001037                      BNE      66$           ;BR IF NOT
(2) (2) 014646 104046                      ERROR   46           ;REPORT DRIVE NOT IN NEUTRAL OR NOT SEIZED
(2) (2) 014650 000137 015034                      JMP     68$           ;BYPASS THE REST OF THE CHECKS
(2) (2) 014654 013737 001170 001126 64$:    MOV      STMP2,$BDDAT  ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
(2) (2) 014662 013737 001226 001234      MOV      PORTB,PTNBR   ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
(2) (2) 014670 113760 001226 000010      MOV      PORTB,RPCS2(RO) ;SELECT PORT B.
(2) (2) 014676 005737 001164                      TST     STMP0         ;SEE IF STATUS EQ 0 FROM PORT A.
(2) (2) 014702 001414                      BEQ     65$           ;BR IF ZERO
(2) (2) 014704 013737 001224 001234      MOV      PORTA,PTNBR   ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
(2) (2) 014712 013737 001172 001126      MOV      STMP3,$BDDAT  ;'BAD DATA' FOR ERROR TYPE OUT
(2) (2) 014720 113760 001224 000010      MOV      PORTA,RPCS2(RO) ;SELECT PORT A.
(2) (2) 014726 005737 001166                      TST     STMP1         ;SEE IF STATUS EQ ZERO FROM PORT B.
(2) (2) 014732 001004                      BNE      66$           ;BR IF NOT
(2) (2) 014734 012737 177777 001250 65$:    MOV      #-1,RELERR   ;SET 'RELEASE ERROR' INDICATOR
(2) (2) 014742 104030                      ERROR   30           ;TYPE ERROR MESSAGE 30
(2) (2) 014744 013737 001170 001126 66$:    MOV      STMP2,$BDDAT  ;LOOK FOR BIT FAILURES WHEN RPDS1 READ
(2) (2) 014752 013737 001224 001234      MOV      PORTA,PTNBR   ;CHANGE PORT NUMBER
(2) (2) 014760 042737 100000 001170      BIC      #ATA,STMP2   ;DON'T CHECK THE ATTN BIT
(2) (2) 014766 023737 001124 001170      CMP      $GDDAT,STMP2 ;ALL BITS OK ?
(2) (2) 014774 001401                      BEQ     67$           ;BR IF OK FROM PORT A.
(2) (2) 014776 104007                      ERROR   7            ;REPORT ERROR
(2) (2) 015000 013737 001172 001126 67$:    MOV      STMP3,$BDDAT  ;CHECK RPDS1 FOR BIT FAILURES - FROM PORT B.
(2) (2) 015006 013737 001226 001234      MOV      PORTB,PTNBR   ;CHANGE PORT NUMBER
(2) (2) 015014 042737 100000 001172      BIC      #ATA,STMP3   ;DON'T CHECK THE ATTN BIT
(2) (2) 015022 023737 001124 001172      CMP      $GDDAT,STMP3 ;SEE IF READ OK FROM PORT B.
(2) (2) 015030 001401                      BEQ     68$           ;BR IF OK
(2) (2) 015032 104007                      ERROR   7            ;REPORT ERROR
(1) (2) 015034 000240 68$:    NOP
(1) (1) 015036 000004                      SCOPE
;LOOP ?

```

8296
8318
8319

```

(3)
(4)
(4)
(4)
(4)
(4)
(4)
(4)
(4)
(4)
(4)
(4)
(4)
(4)
(4)
(3)
(2) 015040
(3) 015040 005737 001274
(3) 015044 001406
(3) 015046 100002
(3) 015050 000137 002622
(3) 015054 012737 177777 001274
(3) 015062 112737 000012 001102
(3) 015070 012737 015112 001106
(3) 015076 012737 015112 001110
(1) 015104 012737 007640 001176
8320 015112 012706 001100
8354
(2)
(2)
(2)
(2)
(2) 015116 113760 001224 000010
(2) 015124 013737 001224 001236
(2) 015132 005060 000012
(3) 015136 113760 001226 000010
(3) 015144 013737 001226 001234
(2) 015152 013737 001226 001240
(2) 015160 016037 000012 001126
(2) 015166 010037 001122
(2) 015172 062737 000012 001122
(2) 015200 005037 001124
(2) 015204 023737 001124 001126
(2) 015212 001403
(2) 015214 104004
(2) 015216 000137 016414
(2) 015222
(3) 015222 113760 001224 000010
(3) 015230 013737 001224 001234
(2) 015236 016037 000012 001126
(2) 015244 012737 011700 001124
(2) 015252 013737 001124 001166
(2) 015260 005137 001166
(2) 015264 013737 001126 001164

```

```

*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 RPS1 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 B: IS SET.
*
*****

```

```

TST12:
TST KYBCTL ;PERFORMING ONLY SINGLE TESTS ?
BEQ 2$ ;BR IF NOT
BPL 1$ ;BR IF JUST ENTERED TEST
JMP EXEC ;RETURN & GET NEXT TEST NUMBER
1$: MOV #-1,KYBCTL ;SET SINGLE TEST INDICATOR
2$: MOVB #12,$STNM ;TEST NUMBER
MOV #TEST12,$LPADR ;LOAD LOOP ON TEST ADDRESS
MOV #TEST12,$LPERR ;LOAD LOOP ON ERROR ADDRESS
MOV #4000,$TIMES ;DO 4000. ITERATIONS
TEST12: MOV #STACK,$SP ;LOAD THE STACK POINTER

```

```

*****
;SEIZE THE DRIVE THROUGH PORT A
MOVB PORTA,RPCS2(RO) ;SELECT PORT A
MOV PORTA,SEIZPT ;STORE SEIZING PORT'S ADDRESS
CLR RPS1(RO) ;WRITE RPS1
MOVB PORTB,RPCS2(RO) ;SELECT PORT B
MOV PORTB,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
MOV PORTB,OPPRT ;'OPPOSITE' PORT ADDRESS
MOV RPS1(RO),$BDDAT ;SEE IF DRIVE SEIZED BY PORT A
MOV RO,$BDAOR ;RH11 BASE ADDRESS
ADD #RPS1,$BDAOR ;GENERATE BAD REGISTER ADDRESS
CLR $GDDAT ;REGISTER SHOULD BE ZERO
CMP $GDDAT,$BDDAT ;IS THE REGISTER ZERO
BEQ 64$ ;BR IF IT IS
ERROR 4 ;REPORT THE ERROR
JMP 1$ ;BYPASS REST OF THE SUBTEST
64$:
MOVB PORTA,RPCS2(RO) ;SELECT PORT A
MOV PORTA,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
MOV RPS1(RO),$BDDAT ;SEE IF SEIZING PORT SEES CORRECT STATUS
MOV #MOL!PGM!DPR!DRY!VV,$GDDAT ;EXPECTED STATUS
MOV $GDDAT,$TMP1 ;USE GOOD DATA AS A MASK
COM $TMP1 ;COMPLEMENT THE EXPECTED STATUS
MOV $BDDAT,$TMP0 ;SAVE THE ACTUAL STATUS

```

```

(2) 015272 043737 001166 001164      BIC      $TMP1,$TMP0      ;CLEAR UNWANTED BITS
(2) 015300 023737 001124 001164      CMP      $GDDAT,$TMP0    ;ARE THE EXPECTED STATUS BITS SET ?
(2) 015306 001401                      BEQ      65$              ;BR IF THEY ARE
(2) 015310 104005                      ERROR 5                    ;REPORT THE ERROR
(2) 015312 000240      65$:      NOP

;*****
;DRIVE CLEAR THROUGH PORT A FIRST
(1)
(1) 015314 012760 000011 000000      MOV      #11,RPCS1(RO)   ;ISSUE DRIVE CLEAR THROUGH PORT A
(1)
(2)
(1)
(1)
(2) 015322 113760 001226 000010      MOV      PORTB,RPCS2(RO) ;SELECT PORT B
(2) 015330 013737 001226 001234      MOV      PORTB,PTNBR    ;MOVE PORT ADDRESS TO LOCAT:JN FOR TYPEOUT
(2) 015336 005037 001244                      CLR      CKERR          ;CLEAR THE 'CHECK ERROR' INDICATOR
(2) 015342 016037 000012 001126      MOV      RPDS1(RO), $BDDAT ;GET CONTENTS OF RPDS1
(2) 015350 012737 000012 001122      MOV      #RPDS1,$B0ADR   ;FORM REGISTER ADDRESS OF ERROR MESSAGE
(2) 015356 060037 001122                      ADD      RO,$B0ADR      ;ADD RH11 BASE ADDRESS
(2) 015362 005037 001124                      CLR      $GDDAT        ;WHAT REGISTER SHOULD BE
(2) 015366 013737 001126 001164      MOV      $BDDAT,$TMP0   ;MOVE REGISTER CONTENTS TO '$TMP0'
(2) 015374 042737 100000 001164      BIC      #C77777,$TMP0  ;SAVE SPECIFIED BITS
(2) 015402 023737 001124 001164      CMP      $GDDAT,$TMP0   ;COMPARE THE BITS
(2) 015410 001414                      BEQ      66$            ;BR IF OK
(2) 015412 013737 001126 001174      MOV      $BDDAT,$TMP4   ;COPY 'BAD DATA'
(2) 015420 042737 077777 001174      BIC      #77777,$TMP4   ;CLEAR THE MASKED BITS
(2) 015426 053737 001174 001124      BIS      $TMP4,$GDDAT   ;'OR' WITH GOOD DATA FOR TYPEOUT
(2) 015434 104033                      ERROR 33                ;TYPE MESSAGE 33
(2) 015436 005137 001244                      COM      CKERR          ;SET THE REGISTER COMPARE ERROR INDICATOR
(2) 015442 000240      66$:      NOP
(2) 015444 113760 001224 000010      MOV      PORTA,RPCS2(RO) ;SELECT PORT A
(2) 015452 013737 001224 001234      MOV      PORTA,PTNBR    ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
(2) 015460 005037 001244                      CLR      CKERR          ;CLEAR THE 'CHECK ERROR' INDICATOR
(2) 015464 016037 000012 001126      MOV      RPDS1(RO), $BDDAT ;GET CONTENTS OF RPDS1
(2) 015472 012737 000012 001122      MOV      #RPDS1,$B0ADR   ;FORM REGISTER ADDRESS OF ERROR MESSAGE
(2) 015500 060037 001122                      ADD      RO,$B0ADR      ;ADD RH11 BASE ADDRESS
(2) 015504 012737 011700 001124      MOV      #MOL!PGM!DPR!DRY!VV,$GDDAT ;WHAT REGISTER SHOULD BE
(2) 015512 013737 001126 001164      MOV      $BDDAT,$TMP0   ;MOVE REGISTER CONTENTS TO '$TMP0'
(2) 015520 042737 100000 001164      BIC      #C77777,$TMP0  ;SAVE SPECIFIED BITS
(2) 015526 023737 001124 001164      CMP      $GDDAT,$TMP0   ;COMPARE THE BITS
(2) 015534 001414                      BEQ      68$            ;BR IF OK
(2) 015536 013737 001126 001174      MOV      $BDDAT,$TMP4   ;COPY 'BAD DATA'
(2) 015544 042737 077777 001174      BIC      #77777,$TMP4   ;CLEAR THE MASKED BITS
(2) 015552 053737 001174 001124      BIS      $TMP4,$GDDAT   ;'OR' WITH GOOD DATA FOR TYPEOUT
(2) 015560 104033                      ERROR 33                ;TYPE MESSAGE 33
(2) 015562 005137 001244                      COM      CKERR          ;SET THE REGISTER COMPARE ERROR INDICATOR
(2) 015566 000240      68$:      NOP

;*****
;NOW ISSUE MASSBUS INIT
(1)
(1) 015570 012760 000040 000010      MOV      #CLR,RPCS2(RO) ;ISSUE MASSBUS INIT
(1)
(2)
(1)
(1)
(2)
;*****
;CONFIRM THAT DRIVE STILL SEIZED BY PORT A

```

E06

```

(1)
(2) 015576 113760 001226 000010      MOVB   PORTB,RPCS2(RO) ;SELECT PORT B
(2) 015604 013737 001226 001234      MOV    PORTB,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
(2) 015612 005037 001244      CLR    CKERR ;CLEAR THE 'CHECK ERROR' INDICATOR
(2) 015616 016037 000012 001126      MOV    RPDS1(RO),SBDDAT ;GET CONTENTS OF RPDS1
(2) 015624 012737 000012 001122      MOV    #RPDS1,SBADR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
(2) 015632 060037 001122      ADD    RO,SBADR ;ADD RH11 BASE ADDRESS
(2) 015636 005037 001124      CLR    $GDDAT ;WHAT REGISTER SHOULD BE
(2) 015642 013737 001126 001164      MOV    SBDDAT,$TMP0 ;MOVE REGISTER CONTENTS TO '$TMP0'
(2) 015650 042737 100000 001164      BIC    #1C7777,$TMP0 ;SAVE SPECIFIED BITS
(2) 015656 023737 001124 001164      CMP    $GDDAT,$TMP0 ;COMPARE THE BITS
(2) 015664 001414      BEQ    70$ ;BR IF OK
(2) 015666 013737 001126 001174      MOV    SBDDAT,$TMP4 ;COPY 'BAD DATA'
(2) 015674 042737 077777 001174      BIC    #77777,$TMP4 ;CLEAR THE MASKED BITS
(2) 015702 053737 001174 001124      BIS    $TMP4,$GDDAT ;'OR' WITH GOOD DATA FOR TYPEOUT
(2) 015710 104034      ERROR  34 ;TYPE MESSAGE 34
(2) 015712 005137 001244      COM    CKERR ;SET THE REGISTER COMPARE ERROR INDICATOR
(2) 015716 000240      NOP
(2) 015720 113760 001224 000010      MOVB   PORTA,RPCS2(RO) ;SELECT PORT A
(2) 015726 013737 001224 001234      MOV    PORTA,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
(2) 015734 005037 001244      CLR    CKERR ;CLEAR THE 'CHECK ERROR' INDICATOR
(2) 015740 016037 000012 001126      MOV    RPDS1(RO),SBDDAT ;GET CONTENTS OF RPDS1
(2) 015746 012737 000012 001122      MOV    #RPDS1,SBADR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
(2) 015754 060037 001122      ADD    RO,SBADR ;ADD RH11 BASE ADDRESS
(2) 015760 012737 011700 001124      MOV    #MOL!PGM!DPR!DRY!VV,$GDDAT ;WHAT REGISTER SHOULD BE
(2) 015766 013737 001126 001164      MOV    SBDDAT,$TMP0 ;MOVE REGISTER CONTENTS TO '$TMP0'
(2) 015774 042737 100000 001164      BIC    #1C7777,$TMP0 ;SAVE SPECIFIED BITS
(2) 016002 023737 001124 001164      CMP    $GDDAT,$TMP0 ;COMPARE THE BITS
(2) 016010 001414      BEQ    72$ ;BR IF OK
(2) 016012 013737 001126 001174      MOV    SBDDAT,$TMP4 ;COPY 'BAD DATA'
(2) 016020 042737 077777 001174      BIC    #77777,$TMP4 ;CLEAR THE MASKED BITS
(2) 016026 053737 001174 001124      BIS    $TMP4,$GDDAT ;'OR' WITH GOOD DATA FOR TYPEOUT
(2) 016034 104034      ERROR  34 ;TYPE MESSAGE 34
(2) 016036 005137 001244      COM    CKERR ;SET THE REGISTER COMPARE ERROR INDICATOR
(2) 016042 000240      NOP
(2)
(2)
(2) ;RELEASE THE DRIVE FROM PORT A
(3) 016044 113760 001224 000010      MOVB   PORTA,RPCS2(RO) ;SELECT PORT A
(3) 016052 013737 001224 001234      MOV    PORTA,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
(3) 016060 012760 000013 000000      MOV    #13,RPDS1(RO) ;ISSUE RELEASE THROUGH PORT A
(3)
(3) ;VERIFY THAT THE DRIVE IS IN NEUTRAL
(3) 016066 005037 001250      CLR    RELERR ;CLEAR THE 'RELEASE ERROR' INDICATOR
(3) 016072 012737 000012 001122      MOV    #RPDS1,SBADR ;FORM THE ADDRESS OF RPDS1 FOR TYPEOUT
(3) 016100 060037 001122      ADD    RO,SBADR ;ADD THE I/O BASE ADDRESS
(3) 016104 012737 011700 001124      MOV    #MOL!PGM!DPR!DRY!VV,$GDDAT ;COMPARISON CONSTANT
(3) 016112 113760 001224 000010      MOVB   PORTA,RPCS2(RO) ;SELECT PORT A.
(3) 016120 016037 000012 001170      MOV    RPDS1(RO),$TMP2 ;GET THE DRIVE STATUS REGISTER FROM PORT A.
(3) 016126 013737 001170 001164      MOV    $TMP2,$TMP0 ;COPY IT INTO '$TMP0'
(3) 016134 042737 100100 001164      BIC    #ATA!VV,$TMP0 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
(3) 016142 113760 001226 000010      MOVB   PORTB,RPCS2(RO) ;SELECT PORT B.
(3) 016150 016037 000012 001172      MOV    RPDS1(RO),$TMP3 ;GET THE DRIVE STATUS REGISTER FROM PORT B.
(3) 016156 013737 001172 001166      MOV    $TMP3,$TMP1 ;COPY IT INTO '$TMP1'
(3) 016164 042737 100100 001166      BIC    #ATA!VV,$TMP1 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
    
```

```

(3) 016172 023737 001164 001166      CMP      $TMP0,$TMP1      ; IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
(3) 016200 001006                      BNE      74$              ; BR IF NOT
(3) 016202 005737 001164                      TST      $TMP0            ; REGISTERS ARE THE SAME: ARE THEY ZERO ?
(3) 016206 001045                      BNE      76$              ; BR IF NOT
(3) 016210 104046                      ERROR    46              ; REPORT DRIVE NOT IN NEUTRAL OR NOT SEIZED
(3) 016212 000137 016412                      JMP      78$              ; BYPASS THE REST OF THE CHECKS
(3) 016216 013737 001170 001126 74$:      MOV      $TMP2,$BDDAT     ; SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
(3) 016224 013737 001226 001234      MOV      PORTB,PTNBR     ; SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
(3) 016232 113760 001226 000010      MOVVB   PORTB,RPCS2(RO) ; SELECT PORT B.
(3) 016240 005737 001164                      TST      $TMP0            ; SEE IF STATUS EQ 0 FROM PORT A.
(3) 016244 001414                      BEQ      75$              ; BR IF ZERO
(3) 016246 013737 001224 001234      MOV      PORTA,PTNBR     ; SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
(3) 016254 013737 001172 001126      MOV      $TMP3,$BDDAT     ; 'BAD DATA' FOR ERROR TYPE OUT
(3) 016262 113760 001224 000010      MOVVB   PORTA,RPCS2(RO) ; SELECT PORT A.
(3) 016270 005737 001166                      TST      $TMP1            ; SEE IF STATUS EQ ZERO FROM PORT B.
(3) 016274 001012                      BNE      76$              ; BR IF NOT
(3) 016276 012737 177777 001250 75$:      MOV      #-1,RELERR      ; SET 'RELEASE ERROR' INDICATOR
(3) 016304 012760 000011 000000      MOV      #11,RPCS1(RO)   ; CLEAR THE DRIVE
(3) 016312 012760 000013 000000      MOV      #13,RPCS1(RO)   ; RELEASE THE DRIVE
(3) 016320 104026                      ERROR    26              ; TYPE ERROR MESSAGE 26
(3) 016322 013737 001170 001126 76$:      MOV      $TMP2,$BDDAT     ; LOOK FOR BIT FAILURES WHEN RPDS1 READ
(3) 016330 013737 001224 001234      MOV      PORTA,PTNBR     ; CHANGE PORT NUMBER
(3) 016336 042737 100000 001170      BIC      #ATA,$TMP2      ; DON'T CHECK THE ATTN BIT
(3) 016344 023737 001124 001170      CMP      $GDDAT,$TMP2    ; ALL BITS OK ?
(3) 016352 001401                      BEQ      77$              ; BR IF OK FROM PORT A.
(3) 016354 104007                      ERROR    7               ; REPORT ERROR
(3) 016356 013737 001172 001126 77$:      MOV      $TMP3,$BDDAT     ; CHECK RPDS1 FOR BIT FAILURES - FROM PORT B.
(3) 016364 013737 001226 001234      MOV      PORTB,PTNBR     ; CHANGE PORT NUMBER
(3) 016372 042737 100000 001172      BIC      #ATA,$TMP3      ; DON'T CHECK THE ATTN BIT
(3) 016400 023737 001124 001172      CMP      $GDDAT,$TMP3    ; SEE IF READ OK FROM PORT B.
(3) 016406 001401                      BEQ      78$              ; BR IF OK
(3) 016410 104007                      ERROR    7               ; REPORT ERROR
(3) 016412 000240 78$:      NOP
(1) 016414 000004 1$:      SCOPE                    ; LOOP ?

```

(1)
(2)
(3)
(4)
(5)
(6)
(7)
(8)
(9)
(10)
(11)
(12)
(13)
(14)
(15)
(16)
(17)
(18)
(19)
(20)

```

*****
*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.
*****
*ST13:      TST      KYBCTL      ;PERFORMING ONLY SINGLE TESTS

```

```

(3) 016416
(3) 016416 005737 001274

```

```

(3) 016422 001406 BEQ 25 ;BR IF NOT
(3) 016424 100002 BPL 15 ;BR IF JUST ENTERED TEST
(3) 016426 000137 002622 JMP EXEC ;RETURN & GET NEXT TEST NUMBER
(3) 016432 012737 177777 001274 15: MOV #-1,KYBCTL ;SET SINGLE TEST INDICATOR
(3) 016440 112737 000013 001102 25: MOVB #13,$STNM ;TEST NUMBER
(3) 016446 012737 016470 001106 MOV #TEST13,$LPADR ;LOAD LOOP ON TEST ADDRESS
(3) 016454 012737 016470 001110 MOV #TEST13,$LPERR ;LOAD LOOP ON ERROR ADDRESS
(1) 016462 012737 007640 001176 MOV #4000, $TIMES ;DO 4000. ITERATIONS
8375 016470 012706 001100 TEST13: MOV #STACK,SP ;LOAD THE STACK POINTER
8376

;*****
;SEIZE THE DRIVE THROUGH PORT B
(2) 016474 113760 001226 000010 MOVB PORTB,RPCS2(RO) ;SELECT PORT B
(2) 016502 013737 001226 001236 MOV PORTB,SEIZPT ;STORE SEIZING PORT'S ADDRESS
(2) 016510 005060 000012 CLR RPDS1(RO) ;WRITE RPDS1
(3) 016514 113760 001224 000010 MOVB PORTA,RPCS2(RO) ;SELECT PORT A
(3) 016522 013737 001224 001234 MOV PORTA,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
(2) 016530 013737 001224 001240 MOV PORTA,OPPRT ;'OPPOSITE' PORT ADDRESS
(2) 016536 016037 000012 001126 MOV RPDS1(RO), $BDDAT ;SEE IF DRIVE SEIZED BY PORT B
(2) 016544 010037 001122 RO,$BDAADR ;R#11 BASE ADDRESS
(2) 016550 062737 000012 001122 ADD #RPDS1,$BDAADR ;GENERATE BAD REGISTER ADDRESS
(2) 016556 005037 001124 CLR $GDDAT ;REGISTER SHOULD BE ZERO
(2) 016562 023737 001124 001126 CMP $GDDAT,$BDDAT ;IS THE REGISTER ZERO
(2) 016570 001403 BEQ 645 ;BR IF IT IS
(2) 016572 104004 ERROR 4 ;REPORT THE ERROR
(2) 016574 000137 017772 JMP 15 ;BYPASS REST OF THE SUBTEST
(2) 016600
(3) 016600 113760 001226 000010 645: MOVB PORTB,RPCS2(RO) ;SELECT PORT B
(3) 016606 013737 001226 001234 MOV PORTB,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
(2) 016614 016037 000012 001126 MOV RPDS1(RO), $BDDAT ;SEE IF SEIZING PORT SEES CORRECT STATUS
(2) 016622 012737 011700 001124 #MOL:PGM!DPR!DRY!VV,$GDDAT ;EXPECTED STATUS
(2) 016630 013737 001124 001166 MOV $GDDAT,$TMP1 ;USE GOOD DATA AS A MASK
(2) 016636 005137 001166 COM $TMP1 ;COMPLEMENT THE EXPECTED STATUS
(2) 016642 013737 001126 001164 MOV $BDDAT,$TMP0 ;SAVE THE ACTUAL STATUS
(2) 016650 043737 001166 001164 BIC $TMP1,$TMP0 ;CLEAR UNWANTED BITS
(2) 016656 023737 001124 001164 CMP $GDDAT,$TMP0 ;ARE THE EXPECTED STATUS BITS SET ?
(2) 016664 001401 BEQ 655 ;BR IF THEY ARE
(2) 016666 104005 ERROR 5 ;REPORT THE ERROR
(2) 016670 000240 655: NOP

;*****
;DRIVE CLEAR THROUGH PORT B FIRST
(1) 016672 012760 000011 000000 MOV #11,RPCS1(RO) ;ISSUE DRIVE CLEAR THROUGH PORT B
(2)
(1)
;*****
;VERIFY THAT DRIVE STILL SEIZED BY PORT B
(2) 016700 113760 001224 000010 MOVB PORTA,RPCS2(RO) ;SELECT PORT A
(2) 016706 013737 001224 001234 MOV PORTA,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
(2) 016714 005037 001244 CLR CKERR ;CLEAR THE 'CHECK ERROR' INDICATOR
(2) 016720 016037 000012 001126 MOV RPDS1(RO), $BDDAT ;GET CONTENTS OF RPDS1
(2) 016726 012737 000012 001122 MOV #RPDS1,$BDAADR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
(2) 016734 060037 001122 ADD RO,$BDAADR ;ADD R#11 BASE ADDRESS

```



```

(2) 016740 005037 001124 CLR $GDDAT ;WHAT REGISTER SHOULD BE
(2) 016744 013737 001126 001164 MOV $BDDAT,$TMP0 ;MOVE REGISTER CONTENTS TO '$TMP0'
(2) 016752 042737 100000 001164 BIC #1C7777,$TMP0 ;SAVE SPECIFIED BITS
(2) 016760 023737 001124 001164 CMP $GDDAT,$TMP0 ;COMPARE THE BITS
(2) 016766 001414 BEQ 66$ ;BR IF OK
(2) 016770 013737 001126 001174 MOV $BDDAT,$TMP4 ;COPY 'BAD DATA'
(2) 016776 042737 077777 001174 BIC #7777,$TMP4 ;CLEAR THE MASKED BITS
(2) 017004 053737 001174 001124 BIS $TMP4,$GDDAT ;'OR' WITH GOOD DATA FOR TYPEOUT
(2) 017012 104033 ERROR 33 ;TYPE MESSAGE 33
(2) 017014 005137 001244 COM CKERR ;SET THE REGISTER COMPARE ERROR INDICATOR
(2) 017020 000240 66$: NOP
(2) 017022 113760 001226 000010 MOVB PORTB,$PC52($R0) ;SELECT PORT B
(2) 017030 013737 001226 001234 MOV PORTB,$PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
(2) 017036 005037 001244 CLR CKERR ;CLEAR THE 'CHECK ERROR' INDICATOR
(2) 017042 016037 000012 001126 MOV RPS1($R0),$BDDAT ;GET CONTENTS OF RPS1
(2) 017050 012737 000012 001122 MOV #RPS1,$BDAOR ;FORM REGISTER ADDRESS OF _RROR MESSAGE
(2) 017056 060037 001122 ADD R0,$BDAOR ;ADD RHI1 BASE ADDRESS
(2) 017062 012737 011700 001124 MOV #M0L!PGM!DPR!DRY!VV,$GDDAT ;WHAT REGISTER SHOULD BE
(2) 017070 013737 001126 001164 MOV $BDDAT,$TMP0 ;MOVE REGISTER CONTENTS TO '$TMP0'
(2) 017076 042737 100000 001164 BIC #1C7777,$TMP0 ;SAVE SPECIFIED BITS
(2) 017104 023737 001124 001164 CMP $GDDAT,$TMP0 ;COMPARE THE BITS
(2) 017112 001414 BEQ 68$ ;BR IF OK
(2) 017114 013737 001126 001174 MOV $BDDAT,$TMP4 ;COPY 'BAD DATA'
(2) 017122 042737 077777 001174 BIC #7777,$TMP4 ;CLEAR THE MASKED BITS
(2) 017130 053737 001174 001124 BIS $TMP4,$GDDAT ;'OR' WITH GOOD DATA FOR TYPEOUT
(2) 017136 104033 ERROR 33 ;TYPE MESSAGE 33
(2) 017140 005137 001244 COM CKERR ;SET THE REGISTER COMPARE ERROR INDICATOR
(2) 017144 000240 68$: NOP

;*****
;NOW ISSUE MASSBUS INIT
017146 012760 000040 000010 MOV #CLR,$PC52($R0) ;ISSUE MASSBUS INIT

;*****
;CONFIRM THAT DRIVE STILL SEIZED BY PORT B
(2) 017154 113760 001224 000010 MOVB PORTA,$PC52($R0) ;SELECT PORT A
(2) 017162 013737 001224 001234 MOV PORTA,$PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
(2) 017170 005037 001244 CLR CKERR ;CLEAR THE 'CHECK ERROR' INDICATOR
(2) 017174 016037 000012 001126 MOV RPS1($R0),$BDDAT ;GET CONTENTS OF RPS1
(2) 017202 012737 000012 001122 MOV #RPS1,$BDAOR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
(2) 017210 060037 001122 ADD R0,$BDAOR ;ADD RHI1 BASE ADDRESS
(2) 017214 005037 001124 CLR $GDDAT ;WHAT REGISTER SHOULD BE
(2) 017220 013737 001126 001164 MOV $BDDAT,$TMP0 ;MOVE REGISTER CONTENTS TO '$TMP0'
(2) 017226 042737 100000 001164 BIC #1C7777,$TMP0 ;SAVE SPECIFIED BITS
(2) 017234 023737 001124 001164 CMP $GDDAT,$TMP0 ;COMPARE THE BITS
(2) 017242 001414 BEQ 70$ ;BR IF OK
(2) 017244 013737 001126 001174 MOV $BDDAT,$TMP4 ;COPY 'BAD DATA'
(2) 017252 042737 077777 001174 BIC #7777,$TMP4 ;CLEAR THE MASKED BITS
(2) 017260 053737 001174 001124 BIS $TMP4,$GDDAT ;'OR' WITH GOOD DATA FOR TYPEOUT
(2) 017266 104034 ERROR 34 ;TYPE MESSAGE 34
(2) 017270 005137 001244 COM CKERR ;SET THE REGISTER COMPARE ERROR INDICATOR
(2) 017274 000240 70$: NOP
(2) 017276 113760 001226 000010 MOVB PORTB,$PC52($R0) ;SELECT PORT B
(2) 017304 013737 001226 001234 MOV PORTB,$PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR T FEOUT

```

```

( ) 017312 005037 001244 CLR CKERR ;CLEAR THE 'CHECK ERROR' INDICATOR
( ) 017316 016037 000012 001126 MOV RPDS1(RO) $BDDAT ;GET CONTENTS OF RPDS1
( ) 017324 012737 000012 001122 MOV #RPDS1,$B0ADR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
( ) 017332 060037 001122 ADD RO,$B0ADR ;ADD RH11 BASE ADDRESS
( ) 017336 012737 011700 001124 MOV #MOL:PGM:DPR:DRY:VV,$GDDAT ;WHAT REGISTER SHOULD BE
( ) 017344 013737 001126 001164 MOV $BDDAT,$TMP0 ;MOVE REGISTER CONTENTS TO 'STMP0'
( ) 017352 042737 100000 001164 BIC #1C7777,$TMP0 ;SAVE SPECIFIED BITS
( ) 017360 023737 001124 001164 CMP $GDDAT,$TMP0 ;COMPARE THE BITS
( ) 017366 001414 BEQ 72$ ;BR IF OK
( ) 017370 013737 001126 001174 MOV $BDDAT,$TMP4 ;COPY 'BAD DATA'
( ) 017376 042737 077777 001174 BIC #7777,$TMP4 ;CLEAR THE MASKED BITS
( ) 017404 053737 001174 001124 BIS $TMP4,$GDDAT ;'OR' WITH GOOD DATA FOR TYPEOUT
( ) 017412 104034 ERROR 34 ;TYPE MESSAGE 34
( ) 017414 005137 001244 COM CKERR ;SET THE REGISTER COMPARE ERROR INDICATOR
( ) 017420 000240 72$: NOP

;RELEASE THE DRIVE FROM PORT B

( ) 017422 113760 001226 000010 MOVB PORTB,RPCS2(RO) ;SELECT PORT B
( ) 017430 013737 001226 001234 MOV PORTB,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
( ) 017436 012760 000013 000000 MOV #13,RPCS1(RO) ;ISSUE RELEASE THROUGH PORT B

;VERIFY THAT THE DRIVE IS IN NEUTRAL

( ) 017444 005037 001250 CLR RELERR ;CLEAR THE 'RELEASE ERROR' INDICATOR
( ) 017450 012737 000012 001122 MOV #RPDS1,$B0ADR ;FORM THE ADDRESS OF RPDS1 FOR TYPEOUT
( ) 017456 060037 001122 ADD RO,$B0ADR ;ADD THE I/O BASE ADDRESS
( ) 017462 012737 011700 001124 MOV #MOL:PGM:DPR:DRY:VV,$GDDAT ;COMPARISON CONSTANT
( ) 017470 113760 001224 000010 MOVB PORTA,RPCS2(RO) ;SELECT PORT 'A'
( ) 017476 016037 000012 001170 MOV RPDS1(RO) $TMP2 ;GET THE DRIVE STATUS REGISTER FROM PORT A.
( ) 017504 013737 001170 001164 MOV $TMP2,$TMP0 ;COPY IT INTO 'STMP0'
( ) 017512 042737 100100 001164 BIC #ATA!VV,$TMP0 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
( ) 017520 113760 001226 000010 MOVB PORTB,RPCS2(RO) ;SELECT PORT B.
( ) 017526 016037 000012 001172 MOV RPDS1(RO) $TMP3 ;GET THE DRIVE STATUS REGISTER FROM PORT B.
( ) 017534 013737 001172 001166 MOV $TMP3,$TMP1 ;COPY IT INTO 'STMP1'
( ) 017542 042737 100100 001166 BIC #ATA!VV,$TMP1 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
( ) 017550 023737 001164 001166 CMP $TMP0,$TMP1 ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS
( ) 017556 001006 BNE 74$ ;BR IF NOT
( ) 017560 005737 001164 TST $TMP0 ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
( ) 017564 001045 BNE 76$ ;BR IF NOT
( ) 017566 104046 ERROR 46 ;REPORT DRIVE NOT IN NEUTRAL OR NOT SEIZED
( ) 017570 000137 017770 JMP 78$ ;BYPASS THE REST OF THE CHECKS
( ) 017574 013737 001170 001126 74$: MOV $TMP2,$BDDAT ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
( ) 017602 013737 001226 001234 MOV PORTB,PTNBR ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
( ) 017610 113760 001226 000010 MOVB PORTB,RPCS2(RO) ;SELECT PORT B.
( ) 017616 005737 001164 TST $TMP0 ;SEE IF STATUS EQ 0 FROM PORT A.
( ) 017622 001414 BEQ 75$ ;BR IF ZERO
( ) 017624 013737 001224 001234 MOV PORTA,PTNBR ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
( ) 017632 013737 001172 001126 MOV $TMP3,$BDDAT ;'BAD DATA' FOR ERROR TYPE OUT
( ) 017640 113760 001224 000010 MOVB PORTA,RPCS2(RO) ;SELECT PORT A.
( ) 017646 005737 001166 TST $TMP1 ;SEE IF STATUS EQ ZERO FROM PORT B.
( ) 017652 001012 BNE 76$ ;BR IF NOT
( ) 017654 012737 177777 001250 75$: MOV #-1,RELERR ;SET 'RELEASE ERROR' INDICATOR
( ) 017662 012760 000011 000000 MOV #11,RPCS1(RO) ;CLEAR THE DRIVE
( ) 017670 012760 000013 000000 MOV #13,RPCS1(RO) ;RELEASE THE DRIVE
( ) 017676 104026 ERROR 26 ;TYPE ERROR MESSAGE 26

```

J06

```

(3) 017700 013737 001170 001126 76$: MOV $TMP2,$BDDAT ;LOOK FOR BIT FAILURES WHEN RPDS1 READ
(3) 017706 013737 001224 001234 ;MOV PORTA,PTNBR ;CHANGE PORT NUMBER
(3) 017714 042737 100000 001170 ;BIC #ATA,$TMP2 ;DON'T CHECK THE ATTN BIT
(3) 017722 023737 001124 001170 ;CMP $GDDAT,$TMP2 ;ALL BITS OK ?
(3) 017730 001401 ;BEQ 77$ ;BR IF OK FROM PORT A.
(3) 017732 104007 ;ERROR 7 ;REPORT ERROR
(3) 017734 013737 001172 001126 77$: MOV $TMP3,$BDDAT ;CHECK RPDS1 FOR BIT FAILURES - FROM PORT B.
(3) 017742 013737 001226 001234 ;MOV PORTB,PTNBR ;CHANGE PORT NUMBER
(3) 017750 042737 100000 001172 ;BIC #ATA,$TMP3 ;DON'T CHECK THE ATTN BIT
(3) 017756 023737 001124 001172 ;CMP $GDDAT,$TMP3 ;SEE IF READ OK FROM PORT B.
(3) 017764 001401 ;BEQ 78$ ;BR IF OK
(3) 017766 104007 ;ERROR 7 ;REPORT ERROR
(3) 017770 000240 78$: NOP
(1) 017772 000004 1$: SCOPE ;LOOP ?

```

00398

```

*****
*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.
*****

```

```

(3) 017774 005737 001274 ;TST KYBCTL ;PERFORMING ONLY SINGLE TESTS ?
(3) 020000 001406 ;BEQ 2$ ;BR IF NOT
(3) 020002 100002 ;BPL 1$ ;BR IF JUST ENTERED TEST
(3) 020004 000137 002622 ;JMP EXEC ;RETURN & GET NEXT TEST NUMBER
(3) 020010 012737 177777 001274 1$: MOV #-1,KYBCTL ;SET SINGLE TEST INDICATOR
(3) 020016 112737 000014 001102 2$: MOV #14,$STNM ;TEST NUMBER
(3) 020024 012737 020046 001106 ;MOV #TEST14,$LPADR ;LOAD LOOP ON TEST ADDRESS
(3) 020032 012737 020046 001110 ;MOV #TEST14,$LPERR ;LOAD LOOP ON ERROR ADDRESS
(1) 020040 012737 000004 001176 ;MOV #4,$TIMES ;DO 4 ITERATIONS
0398 020046 012706 001100 TEST14: MOV #STACK,$SP ;LOAD THE STACK POINTER
0748

```

0398

;SET ATTENTION BITS FOR BOTH PORTS

```

(2) 020052 113760 001224 000010 ;MOVB PORTA,RPCS2(R0) ;SELECT PORT 64$
(2) 020060 012760 177777 000014 ;MOV #-1,RPER1(R0) ;FORCE ERRORS
(2) 020066 005060 000014 ;CLR RPER1(R0) ;CLEAR THE ERRORS
(2) 020072 013760 001226 000010 ;MOV PORTB,RPCS2(R0) ;SELECT THE OTHER PORT
(2) 020100 005760 000012 64$: TST RPDS1(R0) ;WAIT FOR DRIVE TO TIMEOUT
(2) 020104 001775 ;BEQ 64$ ;BR IF DRIVE HASN'T TIMED OUT

```

```

(2) 020106 012760 177777 000014      MOV      #-1,RPER1(RO)  ;FORCE ERRORS ON PORT 65$
(2) 020114 005060 000014      CLR      RPER1(RO)    ;CLEAR THE ERRORS
(2) 020120 113760 001224 000010      MOVVB   PORTA,RPCS2(RO) ;SELECT PORT "64$" AGAIN
(2) 020126 005760 000012      65$:   TST      RPDS1(RO)  ;WAIT FOR DRIVE TO TIMEOUT
(2) 020132 001775      BEQ     65$          ;BR IF DRIVE HASN'T TIMED OUT
(1)
(2)
(1)
(1)
(3) 020134 113760 001224 000010      MOVVB   PORTA,RPCS2(RO) ;SELECT PORT A
(3) 020142 013737 001224 001234      MOV     PORTA,PTNBR   ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
(3) 020150 005037 001244      CLR     CKERR        ;CLEAR THE 'CHECK ERROR' INDICATOR
(3) 020154 016037 000012 001126      MOV     RPDS1(RO), $BDDAT ;GET CONTENTS OF RPDS1
(3) 020162 012737 000012 001122      MOV     #RPDS1,$BDAOR  ;FORM REGISTER ADDRESS OF ERROR MESSAGE
(3) 020170 060037 001122      ADD     RO,$BDAOR     ;ADD RH11 BASE ADDRESS
(3) 020174 012737 100000 001124      MOV     #ATA,$GDDAT   ;WHAT REGISTER SHOULD BE
(3) 020202 013737 001126 001164      MOV     $BDDAT,$TMP0  ;MOVE REGISTER CONTENTS TO '$TMP0'
(3) 020210 042737 077777 001164      BIC     #ICATA,$TMP0  ;SAVE SPECIFIED BITS
(3) 020216 023737 001124 001164      CMP     $GDDAT,$TMP0  ;COMPARE THE BITS
(3) 020224 001414      BEQ     66$          ;BR IF OK
(3) 020226 013737 001126 001174      MOV     $BDDAT,$TMP4  ;COPY 'BAD DATA'
(3) 020234 042737 100000 001174      BIC     #ATA,$TMP4   ;CLEAR THE MASKED BITS
(3) 020242 053737 001174 001124      BIS     $TMP4,$GDDAT  ;'OR' WITH GOOD DATA FOR TYPEOUT
(3) 020250 104010      ERROR  10           ;REPORT THE ERROR
(3) 020252 005137 001244      COM     CKERR        ;SET THE REGISTER COMPARE ERROR INDICATOR
(3) 020256 000240      66$:   NOP
(2) 020260 005737 001244      TST     CKERR        ;WAS ATTN BIT FOR PORT A SET ?
(2) 020264 001402      BEQ     +6           ;BR IF IT WAS
(2) 020266 000137 021274      JMP     1$          ;BYPASS REST OF TEST IF NOT
(3) 020272 113760 001226 000010      MOVVB   PORTB,RPCS2(RO) ;SELECT PORT B
(3) 020300 013737 001226 001234      MOV     PORTB,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
(3) 020306 005037 001244      CLR     CKERR        ;CLEAR THE 'CHECK ERROR' INDICATOR
(3) 020312 016037 000012 001126      MOV     RPDS1(RO), $BDDAT ;GET CONTENTS OF RPDS1
(3) 020320 012737 000012 001122      MOV     #RPDS1,$BDAOR  ;FORM REGISTER ADDRESS OF ERROR MESSAGE
(3) 020326 060037 001122      ADD     RO,$BDAOR     ;ADD RH11 BASE ADDRESS
(3) 020332 012737 100000 001124      MOV     #ATA,$GDDAT   ;WHAT REGISTER SHOULD BE
(3) 020340 013737 001126 001164      MOV     $BDDAT,$TMP0  ;MOVE REGISTER CONTENTS TO '$TMP0'
(3) 020346 042737 077777 001164      BIC     #ICATA,$TMP0  ;SAVE SPECIFIED BITS
(3) 020354 023737 001124 001164      CMP     $GDDAT,$TMP0  ;COMPARE THE BITS
(3) 020362 001414      BEQ     68$          ;BR IF OK
(3) 020364 013737 001126 001174      MOV     $BDDAT,$TMP4  ;COPY 'BAD DATA'
(3) 020372 042737 100000 001174      BIC     #ATA,$TMP4   ;CLEAR THE MASKED BITS
(3) 020400 053737 001174 001124      BIS     $TMP4,$GDDAT  ;'OR' WITH GOOD DATA FOR TYPEOUT
(3) 020406 104010      ERROR  10           ;REPORT THE ERROR
(3) 020410 005137 001244      COM     CKERR        ;SET THE REGISTER COMPARE ERROR INDICATOR
(3) 020414 000240      68$:   NOP
(2) 020416 005737 001244      TST     CKERR        ;WAS ATTN BIT FOR PORT B SET ?
(2) 020422 001402      BEQ     +6           ;BR IF IT WAS
(2) 020424 000137 021274      JMP     1$          ;BYPASS REST OF TEST IF NOT
(1)
(2)
(2)
(2)
(2)
(2)
(2) 020430 113760 001224 000010      MOVVB   PORTA,RPCS2(RO) ;SELECT PORT A
(2) 020436 013737 001224 001236      MOV     PORTA,SEIZPT ;STORE SEIZING PORT'S ADDRESS

```

;CONFIRM THAT BOTH ATTENTION BITS ARE SET

66\$:

68\$:

;SEIZE THE DRIVE THROUGH PORT A

```

(2) 020444 005060 000012          CLR    RPDS1(RO)      ;WRITE RPDS1
(2) 020450 013737 001226 001240  MOV    PORTB,OPRT    ;'OPPOSITE' PORT ADDRESS
(1)                                     ;*****
(1)                                     ;ISSUE MASSBUS INIT TO PORT A
(1) 020456 012760 000040 000010    MOV    #CLK,RPDS2(RO) ;MASSBUS INIT
(1) 020464 113760 001224 000010    MOV    PORTA,RPDS2(RO);SELECT PORT A AGAIN
(2)                                     ;*****
(1)                                     ;VERIFY THAT ATTENTION BIT FOR PORT A CLEARED
(1)
(2) 020472 005037 001244          CLR    CKERR         ;CLEAR THE 'CHECK ERROR' INDICATOR
(2) 020476 016037 000012 001126    MOV    RPDS1(RO),%SDDAT ;GET CONTENTS OF RPDS1
(2) 020504 012737 000012 001122    MOV    #RPDS1,%SDDADR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
(2) 020512 060037 001122          ADD    RO,%SDDADR    ;ADD RH11 BASE ADDRESS
(2) 020516 005037 001124          CLR    %SDDAT        ;WHAT REGISTER SHOULD BE
(2) 020522 013737 001126 001164    MOV    %SDDAT,%STMP0 ;MOVE REGISTER CONTENTS TO 'STMP0'
(2) 020530 042737 077777 001164    BIC    #%CATA,%STMP0 ;SAVE SPECIFIED BITS
(2) 020536 023737 001124 001164    CMP    %SDDAT,%STMP0 ;COMPARE THE BITS
(2) 020544 001414          BEQ    72$          ;BR IF OK
(2) 020546 013737 001126 001174    MOV    %SDDAT,%STMP4 ;COPY 'BAD DATA'
(2) 020554 042737 100000 001174    BIC    #ATA,%STMP4   ;CLEAR THE MASKED BITS
(2) 020562 053737 001174 001124    BIS    %STMP4,%SDDAT ;'OR' WITH GOOD DATA FOR TYPEOUT
(2) 020570 104047          ERROR  47          ;TYPE MESSAGE 47
(2) 020572 005137 001244          COM    CKERR        ;SET THE REGISTER COMPARE ERROR INDICATOR
(2) 020576 000240          72$: NOP
(1)
(2)                                     ;*****
(2)                                     ;RELEASE THE DRIVE FROM PORT A
(3) 020600 113760 001224 000010    MOV    PORTA,RPDS2(RO);SELECT PORT A
(3) 020606 013737 001224 001234    MOV    PORTA,PTNBR   ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
(2) 020614 012760 000013 000000    MOV    #13,RPDS1(RO);ISSUE RELEASE THROUGH PORT A
(3)                                     ;VERIFY THAT THE DRIVE IS IN NEUTRAL
(3)
(3) 020622 005037 001250          CLR    RELERR        ;CLEAR THE 'RELEASE ERROR' INDICATOR
(3) 020626 012737 000012 001122    MOV    #RPDS1,%SDDADR ;FORM THE ADDRESS OF RPDS1 FOR TYPEOUT
(3) 020634 060037 001122          ADD    RO,%SDDADR    ;ADD THE I/O BASE ADDRESS
(3) 020640 012737 011700 001124    MOV    #MOL!PGM!DPR!DRY!VV,%SDDAT ;COMPARISON CONSTANT
(3) 020646 113760 001224 000010    MOV    PORTA,RPDS2(RO);SELECT PORT A.
(3) 020654 016037 000012 001170    MOV    RPDS1(RO),%STMP2 ;GET THE DRIVE STATUS REGISTER FROM PORT A.
(3) 020662 013737 001170 001164    MOV    %STMP2,%STMP0 ;COPY IT INTO 'STMP0'
(3) 020670 042737 100100 001164    BIC    #ATA!VV,%STMP0 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
(3) 020676 113760 001226 000010    MOV    PORTB,RPDS2(RO);SELECT PORT B.
(3) 020704 016037 000012 001172    MOV    RPDS1(RO),%STMP3 ;GET THE DRIVE STATUS REGISTER FROM PORT B.
(2) 020712 013737 001172 001166    MOV    %STMP3,%STMP1 ;COPY IT INTO 'STMP1'
(3) 020720 042737 100100 001166    BIC    #ATA!VV,%STMP1 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
(3) 020726 023737 001164 001166    CMP    %STMP0,%STMP1 ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS
(3) 020734 001006          BNE    74$          ;BR IF NOT
(3) 020736 005737 001164          TST    %STMP0       ;REGISTERS ARE THE SAME: ARE THEY ZERO?
(3) 020742 001045          BNE    76$          ;BR IF NOT
(3) 020744 104046          ERROR  46          ;REPORT DRIVE NOT IN NEUTRAL OR NOT SEIZED
(3) 020746 000137 021146          JMP    78$          ;BYPASS THE REST OF THE CHECKS
    
```

```

(3) 020752 013737 001170 001126 74S: MOV $TMP2,$BDDAT ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
(3) 020760 013737 001226 001234 MOV PORTB,PTNBR ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
(3) 020766 113760 001226 000010 MOVB PORTB,RPCS2(RO) ;SELECT PORT B.
(3) 020774 005737 001164 TST $TMP0 ;SEE IF STATUS EQ 0 FROM PORT A.
(3) 021000 001414 BEQ 75S ;BR IF ZERO
(3) 021002 013737 001224 001234 MOV PORTA,PTNBR ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
(3) 021010 013737 001172 001126 MOV $TMP3,$BDDAT ;'BAD DATA' FOR ERROR TYPE OUT
(3) 021016 113760 001224 000010 MOVB PORTA,RPCS2(RO) ;SELECT PORT A.
(3) 021024 005737 001166 TST $TMP1 ;SEE IF STATUS EQ ZERO FROM PORT B.
(3) 021030 001012 BNE 76S ;BR IF NOT
(3) 021032 012737 177777 001250 75S: MOV #-1,RELERR ;SET 'RELEASE ERROR' INDICATOR
(3) 021040 012760 000011 000000 MOV #11,RPCS1(RO) ;CLEAR THE DRIVE
(3) 021046 012760 000013 000000 MOV #13,RPCS1(RO) ;RELEASE THE DRIVE
(3) 021054 104026 ERROR 26 ;TYPE ERROR MESSAGE 26
(3) 021056 013737 001170 001126 76S: MOV $TMP2,$BDDAT ;LOOK FOR BIT FAILURES WHEN RPOS1 READ
(3) 021064 013737 001224 001234 MOV PORTA,PTNBR ;CHANGE PORT NUMBER
(3) 021072 042737 100000 001170 BIC #ATA,$TMP2 ;DON'T CHECK THE ATTN BIT
(3) 021100 023737 001124 001170 CMP $GDDAT,$TMP2 ;ALL BITS OK ?
(3) 021106 001401 BEQ 77S ;BR IF OK FROM PORT A.
(3) 021110 104007 ERROR 7 ;REPORT ERROR
(3) 021112 013737 001172 001126 77S: MOV $TMP3,$BDDAT ;CHECK RPOS1 FOR BIT FAILURES - FROM PORT B.
(3) 021120 013737 001226 001234 MOV PORTB,PTNBR ;CHANGE PORT NUMBER
(3) 021126 042737 100000 001172 BIC #ATA,$TMP3 ;DON'T CHECK THE ATTN BIT
(3) 021134 023737 001124 001172 CMP $GDDAT,$TMP3 ;SEE IF READ OK FROM PORT B.
(3) 021142 001401 BEQ 78S ;BR IF OK
(3) 021144 104007 ERROR 7 ;REPORT ERROR
(3) 021146 000240 78S: NOP

```

```

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

```

```

(2) 021150 113760 001226 000010 MOVB PORTB,RPCS2(RO) ;SELECT PORT B
(2) 021156 013737 001226 001234 MOV PORTB,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
(2) 021164 005037 001244 CLR CKERR ;CLEAR THE 'CHECK ERROR' INDICATOR
(2) 021170 016037 000012 001126 MOV RPOS1(RO), $BDDAT ;GET CONTENTS OF RPOS1
(2) 021176 012737 000012 001122 MOV #RPOS1,$BDDAR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
(2) 021204 060037 001122 ADD RO,$BDDAR ;ADD RH11 BASE ADDRESS
(2) 021210 012737 100000 001124 MOV #ATA,$GDDAT ;WHAT REGISTER SHOULD BE
(2) 021216 013737 001126 001164 MOV $BDDAT,$TMP0 ;MOVE REGISTER CONTENTS TO '$TMP0'
(2) 021224 042737 077777 001164 BIC #ICATA,$TMP0 ;SAVE SPECIFIED BITS
(2) 021232 023737 001124 001164 CMP $GDDAT,$TMP0 ;COMPARE THE BITS
(2) 021240 001414 BEQ 79S ;BR IF OK
(2) 021242 013737 001126 001174 MOV $BDDAT,$TMP4 ;COPY 'BAD DATA'
(2) 021250 042737 100000 001174 BIC #ATA,$TMP4 ;CLEAR THE MASKED BITS
(2) 021256 053737 001174 001124 BIS $TMP4,$GDDAT ;'OR' WITH GOOD DATA FOR TYPEOUT
(2) 021264 104050 ERROR 50 ;TYPE MESSAGE 50
(2) 021266 005137 001244 COM CKERR ;SET THE REGISTER COMPARE ERROR INDICATOR
(2) 021272 000240 79S: NOP
(1) 021274 000004 1S: SCOPE ;LOOP ?

```

```

;*****
;TEST 15 TEST RESET ATTENTION 'B' BY MASSBUS CLEAR
;
;VERIFY THAT A MASSBUS INITIALIZE CLEARS ONLY THE ATTENTION BIT OF THE
; SEIZING PORT.
;

```

8456
8457
(3)
(4)
(4)
(4)

- * 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 RPDS!.
- * 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.

(4)
(4)
(4)
(4)
(4)
(4)
(4)
(4)
(4)
(4)
(4)
(2)
(2)
(3)
(3)
(3)
(3)
(3)
(3)
(3)
(3)
(3)
(3)
(1)
B458
B459
(2)
(2)
(2)
(2)
(2)
(2)
(2)
(2)
(2)
(2)
(2)
(1)
(2)
(1)
(1)
(1)
(3)
(3)
(3)
(3)
(3)
(3)
(3)
(3)
(3)
(3)
(3)
(3)
(3)
(3)
(3)
(3)
(3)
(3)
(3)
(3)
(3)
(3)
(3)
(3)
(3)
(3)
(3)

```

021276
021276 005737 001274
021302 001406
021304 100002
021306 000137 002622
021312 012737 177777 001274
021320 112737 000015 001102
021326 012737 021350 001106
021334 012737 021350 001110
021342 012737 000004 001176
021350 012706 001100

021354 113760 001224 000010
021362 012760 177777 000014
021370 005060 000014
021374 013760 001226 000010
021402 005760 000012
021406 001775
021410 012760 177777 000014
021416 005060 000014
021422 113760 001224 000010
021430 005760 000012
021434 001775

021436 113760 001226 000010
021444 013737 001226 001234
021452 005037 001244
021456 016037 000012 001126
021464 012737 000012 001122
021472 060037 001122
021476 012737 100000 001124
021504 013737 001126 001164
021512 042737 077777 001164
021520 023737 001124 001164
021526 001414
021530 013737 001126 001174
021536 042737 100000 001174
021544 053737 001174 001124
    
```

```

*****
TST15:
TST      KYBCTL      ;PERFORMING ONLY SINGLE TESTS ?
BEQ      2$          ;BR IF NOT
BPL      1$          ;BR IF JUST ENTERED TEST
JMP      EXEC        ;RETURN & GET NEXT TEST NUMBER
1$:      MOV         #-1,KYBCTL ;SET SINGLE TEST INDICATOR
2$:      MOV         #15,$STSTM ;TEST NUMBER
        MOV         #TEST15,$LPADR ;LOAD LOOP ON TEST ADDRESS
        MOV         #TEST15,$LPERR ;LOAD LOOP ON ERROR ADDRESS
        MOV         #4,$TIMES    ;DO 4 ITERATIONS
TEST15:  MOV         #STACK,$SP  ;LOAD THE STACK POINTER
;*****

;SET ATTENTION BITS FOR BOTH PORTS
        MOV         PORTA,$RPCS2(R0) ;SELECT PORT 64$
        MOV         #-1,$RPER1(R0)  ;FORCE ERRORS
        CLR         $RPER1(R0)      ;CLEAR THE ERRORS
        MOV         PORTB,$RPCS2(R0) ;SELECT THE OTHER PORT
64$:    TST         $RPDS1(R0)      ;WAIT FOR DRIVE TO TIMEOUT
        BEQ         64$            ;BR IF DRIVE HASN'T TIMED OUT
        MOV         #-1,$RPER1(R0)  ;FORCE ERRORS ON PORT 65$
        CLR         $RPER1(R0)      ;CLEAR THE ERRORS
65$:    MOV         PORTA,$RPCS2(R0) ;SELECT PORT "64$" AGAIN
        TST         $RPDS1(R0)      ;WAIT FOR DRIVE TO TIMEOUT
        BEQ         65$            ;BR IF DRIVE HASN'T TIMED OUT

;*****
;CONFIRM THAT BOTH ATTENTION BITS ARE SET
        MOV         PORTB,$RPCS2(R0) ;SELECT PORT B
        MOV         PORTB,$PTNBR    ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
        CLR         $CKERR          ;CLEAR THE 'CHECK ERROR' INDICATOR
        MOV         $RPDS1(R0),$SBDDAT ;GET CONTENTS OF RPDS1
        MOV         #RPS1,$SBADR    ;FORM REGISTER ADDRESS OF ERROR MESSAGE
        ADD         $R0,$SBADR      ;ADD RHI1 BASE ADDRESS
        MOV         #ATA,$GDDAT     ;WHAT REGISTER SHOULD BE
        MOV         $SBDDAT,$STMP0  ;MOVE REGISTER CONTENTS TO '$STMP0'
        BIC         #1CATA,$STMP0  ;SAVE SPECIFIED BITS
        CMP         $GDDAT,$STMP0  ;COMPARE THE BITS
        BEQ         66$            ;BR IF OK
        MOV         $SBDDAT,$STMP4  ;COPY 'BAD DATA'
        BIC         #ATA,$STMP4    ;CLEAR THE MASKED BITS
        BIS         $STMP4,$GDDAT   ;'OR' WITH GOOD DATA FOR TYPEOUT
    
```

```

(3) 021552 104010          ERROR 10          ;REPORT THE ERROR
(3) 021554 005137 001244    COM      CKERR          ;SET THE REGISTER COMPARE ERROR INDICATOR
(3) 021560 000240          66$:  NOP          ;
(2) 021562 005737 001244    TST     CKERR          ;WAS ATTN BIT FOR PORT B SET ?
(2) 021566 001402          BEQ     .+6           ;BR IF IT WAS
(2) 021570 000137 022576    JMP     IS           ;BYPASS REST OF TEST IF NOT
(3) 021574 113760 001224 000010  MOVB   PORTA,RPCS2(RO) ;SELECT PORT A
(3) 021602 013737 001224 001234  MOV    PORTA,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
(3) 021610 005037 001244    CLR    CKERR        ;CLEAR THE 'CHECK ERROR' INDICATOR
(3) 021614 016037 000012 001126  MOV    RPDS1(RO),SBDDAT ;GET CONTENTS OF RPDS1
(3) 021622 012737 000012 001122  MOV    #RPDS1,SBDAOR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
(3) 021530 060037 001122    ADD    RO,SBDAOR    ;ADD RH11 BASE ADDRESS
(3) 021634 012737 100000 001124  MOV    #ATA,$GDDAT ;WHAT REGISTER SHOULD BE
(3) 021642 013737 001126 001164  MOV    SBDDAT,$TMPO ;MOVE REGISTER CONTENTS TO '$TMPO'
(3) 021650 042737 077777 001164  BIC    #ICATA,$TMPO ;SAVE SPECIFIED BITS
(3) 021656 023737 001124 001154  CMP    $GDDAT,$TMPO ;COMPARE THE BITS
(3) 021664 001414          BEQ     68$         ;BR IF OK
(3) 021666 013737 001126 001174  MOV    SBDDAT,$TMP4 ;COPY 'BAD DATA'
(3) 021674 042737 100000 001174  BIC    #ATA,$TMP4   ;CLEAR THE MASKED BITS
(3) 021702 053737 001174 001124  BIS    $TMP4,$GDDAT ;'OR' WITH GOOD DATA FOR TYPEOUT
(3) 021710 104010          ERROR 10          ;REPORT THE ERROR
(3) 021712 005137 001244    COM      CKERR          ;SET THE REGISTER COMPARE ERROR INDICATOR
(3) 021716 000240          68$:  NOP          ;
(2) 021720 005737 001244    TST     CKERR          ;WAS ATTN BIT FOR PORT A SET ?
(2) 021724 001402          BEQ     .+6           ;BR IF IT WAS
(2) 021726 000137 022576    JMP     IS           ;BYPASS REST OF TEST IF NOT
(1)
(2)
(2)
(2)
(2)
(2) 021732 113760 001226 000010  MOVB   PORTB,RPCS2(RO) ;SELECT PORT B
(2) 021740 013737 001226 001236  MOV    PORTB,SEIZPT ;STORE SEIZING PORT'S ADDRESS
(2) 021746 005060 000012          CLR    RPDS1(RO)    ;WRITE RPDS1
(2) 021752 013737 001224 001240  MOV    PORTA,OPPRT  ;'OPPOSITE' PORT ADDRESS
(1)
(2)
(1)
(1)
(1) 021760 012760 000040 000010  MOV    #CLR,RPCS2(RO) ;MASSBUS INIT
(1) 021766 113760 001226 000010  MOVB   PORTB,RPCS2(RO) ;SELECT PORT B AGAIN
(1)
(2)
(1)
(1)
(1)
(2) 021774 005037 001244          CLR    CKERR        ;CLEAR THE 'CHECK ERROR' INDICATOR
(2) 022000 016037 000012 001126  MOV    RPDS1(RO),SBDDAT ;GET CONTENTS OF RPDS1
(2) 022006 012737 000012 001122  MOV    #RPDS1,SBDAOR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
(2) 022014 060037 001122    ADD    RO,SBDAOR    ;ADD RH11 BASE ADDRESS
(2) 022020 005037 001124    CLR    $GDDAT       ;WHAT REGISTER SHOULD BE
(2) 022024 013737 001126 001164  MOV    SBDDAT,$TMPO ;MOVE REGISTER CONTENTS TO '$TMPO'
(2) 022032 042737 077777 001164  BIC    #ICATA,$TMPO ;SAVE SPECIFIED BITS
(2) 022040 023737 001124 001164  CMP    $GDDAT,$TMPO ;COMPARE THE BITS
(2) 022046 001414          BEQ     72$         ;BR IF OK
(2) 022050 013737 001126 001174  MOV    SBDDAT,$TMP4 ;COPY 'BAD DATA'
(2) 022056 042737 100000 001174  BIC    #ATA,$TMP4   ;CLEAR THE MASKED BITS

```



```

(2) 022064 053737 001174 001124      BIS      $TMP4,$GDDAT      ;'OR' WITH GOOD DATA FOR TYPEOUT
(2) 022072 104047                      ERROR    47                ;TYPE MESSAGE 47
(2) 022074 005137 001244                      COM      CKERR             ;SET THE REGISTER COMPARE ERROR INDICATOR
(2) 022100 000240      72$:      NOP
(1)
(2)
(2)
(2)
(3) 022102 113760 001226 000010      MOV      PORTB,RPCS2(RO)   ;SELECT PORT B
(3) 022110 013737 001226 001234      MOV      PORTB,PTNBR      ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
(2) 022116 012760 000013 000000      MOV      #13,RPCS1(RO)    ;ISSUE RELEASE THROUGH PORT B
(3)
(3)
(3)
(3)
(3) 022124 005037 001250                      CLR      RELERR           ;CLEAR THE 'RELEASE ERROR' INDICATOR
(3) 022130 012737 000012 001122      MOV      #RPDS1,$BDDADR   ;FORM THE ADDRESS OF RPDS1 FOR TYPEOUT
(3) 022136 060037 001122                      ADD      RO,$BDDADR       ;ADD THE I/O BASE ADDRESS
(3) 022142 012737 011700 001124      MOV      #MOL!PGM!DPR!DRY!VV,$GDDAT ;COMPARISON CONSTANT
(3) 022150 113760 001224 000010      MOV      PORTA,RPCS2(RO)  ;SELECT PORT A.
(3) 022156 016037 000012 001170      MOV      RPDS1(RO),$TMP2  ;GET THE DRIVE STATUS REGISTER FROM PORT A.
(3) 022164 013737 001170 001164      MOV      $TMP2,$TMP0      ;COPY IT INTO '$TMP0'
(3) 022172 042737 100100 001164      BIC      #ATA!VV,$TMP0    ;CLEAR PORT DEPENDENT BITS FROM THE COPY
(3) 022200 113760 001226 000010      MOV      PORTB,RPCS2(RO)  ;SELECT PORT B.
(3) 022206 016037 000012 001172      MOV      RPDS1(RO),$TMP3  ;GET THE DRIVE STATUS REGISTER FROM PORT B.
(3) 022214 013737 001172 001166      MOV      $TMP3,$TMP1     ;COPY IT INTO '$TMP1'
(3) 022222 042737 100100 001166      BIC      #ATA!VV,$TMP1    ;CLEAR PORT DEPENDENT BITS FROM THE COPY
(3) 022230 023737 001164 001166      CMP      $TMP0,$TMP1     ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
(3) 022236 001006                      BNE      74$             ;BR IF NOT
(3) 022240 005737 001164                      TST      $TMP0           ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
(3) 022244 001045                      BNE      76$             ;BR IF NOT
(3) 022246 104046                      ERROR    46             ;REPORT DRIVE NOT IN NEUTRAL OR NOT SEIZED
(3) 022250 000137 022450                      JMP      78$             ;BYPASS THE REST OF THE CHECKS
(3) 022254 013737 001170 001126 74$:      MOV      $TMP2,$BDDAT     ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
(3) 022262 013737 001226 001234      MOV      PORTB,PTNBR     ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
(3) 022270 113760 001226 000010      MOV      PORTB,RPCS2(RO) ;SELECT PORT B.
(3) 022276 005737 001164                      TST      $TMP0           ;SEE IF STATUS EQ 0 FROM PORT A.
(3) 022302 001414                      BEQ      75$             ;BR IF ZERO
(3) 022304 013737 001224 001234      MOV      PORTA,PTNBR     ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
(3) 022312 013737 001172 001126      MOV      $TMP3,$BDDAT    ;'BAD DATA' FOR ERROR TYPE OUT
(3) 022320 113760 001224 000010      MOV      PORTA,RPCS2(RO) ;SELECT PORT A.
(3) 022326 005737 001166                      TST      $TMP1           ;SEE IF STATUS EQ ZERO FROM PORT B.
(3) 022332 001012                      BNE      76$             ;BR IF NOT
(3) 022334 012737 177777 001250 75$:      MOV      #-1,RELERR      ;SET 'RELEASE ERROR' INDICATOR
(3) 022342 012760 000011 000000      MOV      #11,RPCS1(RO)   ;CLEAR THE DRIVE
(3) 022350 012760 000013 000000      MOV      #13,RPCS1(RO)   ;RELEASE THE DRIVE
(3) 022356 104026                      ERROR    26             ;TYPE ERROR MESSAGE 26
(3) 022360 013737 001170 001126 76$:      MOV      $TMP2,$BDDAT     ;LOOK FOR BIT FAILURES WHEN RPDS1 READ
(3) 022366 013737 001224 001234      MOV      PORTA,PTNBR     ;CHANGE PORT NUMBER
(3) 022374 042737 100000 001170      BIC      #ATA,$TMP2      ;DON'T CHECK THE ATTN BIT
(3) 022402 023737 001124 001170      CMP      $GDDAT,$TMP2    ;ALL BITS OK ?
(3) 022410 001401                      BEQ      77$             ;BR IF OK FROM PORT A.
(3) 022412 104007                      ERROR    7              ;REPORT ERROR
(3) 022414 013737 001172 001126 77$:      MOV      $TMP3,$BDDAT    ;CHECK RPDS1 FOR BIT FAILURES - FROM PORT B.
(3) 022422 013737 001226 001234      MOV      PORTB,PTNBR     ;CHANGE PORT NUMBER
(3) 022430 042737 100000 001172      BIC      #ATA,$TMP3      ;DON'T CHECK THE ATTN BIT

```

```

(3) 022436 023737 001124 001172    CMP    $GDDAT,$TMP3    ;SEE IF READ OK FROM PORT B.
(3) 022444 001401                    BEQ    78$             ;BR IF OK
(3) 022446 104007                    ERROR  7              ;REPORT ERROR
(3) 022450 000240                    78$:  NOP
(1)
(2)
(1)
(1)
(2) 022452 113760 001224 000010    MOVB   PORTA,RPCS2(RO) ;SELECT PORT A
(2) 022460 013737 001224 001234    MOV    PORTA,PTNBR    ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
(2) 022466 005037 001244                    CLR    CKERR          ;CLEAR THE 'CHECK ERROR' INDICATOR
(2) 022472 016037 000012 001126    MOV    RPOS1(RO),$BDDAT ;GET CONTENTS OF RPOS1
(2) 022500 012737 000012 001122    MOV    #RPOS1,$B0ADR  ;FORM REGISTER ADDRESS OF ERROR MESSAGE
(2) 022506 060037 001122                    ADD    RO,$B0ADR      ;ADD RH11 BASE ADDRESS
(2) 022512 012737 100000 001124    MOV    #ATA,$GDDAT    ;WHAT REGISTER SHOULD BE
(2) 022520 013737 001126 001164    MOV    $BDDAT,$TMP0   ;MOVE REGISTER CONTENTS TO '$TMP0'
(2) 022526 042737 077777 001164    BIC    #ICATA,$TMP0   ;SAVE SPECIFIED BITS
(2) 022534 023737 001124 001164    CMP    $GDDAT,$TMP0   ;COMPARE THE BITS
(2) 022542 001414                    BEQ    79$             ;BR IF OK
(2) 022544 013737 001126 001174    MOV    $BDDAT,$TMP4   ;COPY 'BAD DATA'
(2) 022552 042737 100000 001174    BIC    #ATA,$TMP4     ;CLEAR THE MASKED BITS
(2) 022560 053737 001174 001124    BIS    $TMP4,$GDDAT   ;'OR' WITH GOOD DATA FOR TYPEOUT
(2) 022566 104050                    ERROR  50             ;TYPE MESSAGE 50
(2) 022570 005137 001244                    COM    CKERR          ;SET THE REGISTER COMPARE ERROR INDICATOR
(2) 022574 000240                    79$:  NOP
(1) 022576 000004                    1$:  SCOPE             ;LOOP ?

```

8460
8464
8478
8479
(3)
(4)
(4)
(4)
(4)
(4)
(4)
(4)
(4)
(4)
(4)
(3)
(2)
(3)
(3)
(3)
(3)
(3)
(3)
(3)
(1)
8480
8515
(2)

```

;*****
;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.
;*****
TEST16:
TST    KYBCTL          ;PERFORMING ONLY SINGLE TESTS ?
BEQ    2$              ;BR IF NOT
BPL    1$              ;BR IF JUST ENTERED TEST
JMP    EXEC           ;RETURN & GET NEXT TEST NUMBER
1$:  MOV    #-1,KYBCTL ;SET SINGLE TEST INDICATOR
2$:  MOVB   #16,$STNM  ;TEST NUMBER
      MOV    #TEST16,$LPADR ;LOAD LOOP ON TEST ADDRESS
      MOV    #TEST16,$LPERR ;LOAD LOOP ON ERROR ADDRESS
      MOV    #4,$TIMES  ;DO 4 ITERATIONS
TEST16: MOV    #STACK,$SP ;LOAD THE STACK POINTER
;*****

```

```

(2) ;SET ATTENTION BITS FOR BOTH PORTS
(2)
(2) 022656 113760 001224 000010 MOVB PORTA,RPCS2(RO) ;SELECT PORT 64$
(2) 022664 012760 177777 000014 MOV #-1,RPER1(RO) ;FORCE ERRORS
(2) 022672 005060 000014 CLR RPER1(RO) ;CLEAR THE ERRORS
(2) 022676 013760 001226 000010 MOV PORTB,RPCS2(RO) ;SELECT THE OTHER PORT
(2) 022704 005760 000012 64$: TST RPDS1(RO) ;WAIT FOR DRIVE TO TIMEOUT
(2) 022710 001775 BEQ 64$ ;BR IF DRIVE HASN'T TIMED OUT
(2) 022712 012760 177777 000014 MOV #-1,RPER1(RO) ;FORCE ERRORS ON PORT 65$
(2) 022720 005060 000014 CLR RPER1(RO) ;CLEAR THE ERRORS
(2) 022724 113760 001224 000010 MOVB PORTA,RPCS2(RO) ;SELECT PORT "64$" AGAIN
(2) 022732 005760 000012 65$: TST RPDS1(RO) ;WAIT FOR DRIVE TO TIMEOUT
(2) 022736 001775 BEQ 65$ ;BR IF DRIVE HASN'T TIMED OUT
(1)
(2) ;*****
(1) ;CONFIRM THAT BOTH ATTENTION BITS ARE SET
(1)
(3) 022740 113760 001224 000010 MOVB PORTA,RPCS2(RO) ;SELECT PORT A
(3) 022746 013737 001224 001234 MOV PORTA,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
(3) 022754 005037 001244 CLR CKERR ;CLEAR THE 'CHECK ERROR' INDICATOR
(3) 022760 016037 000012 001126 MOV RPDS1(RO),SBDDAT ;GET CONTENTS OF RPDS1
(3) 022766 012737 000012 001122 MOV #RPDS1,SBDAOR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
(3) 022774 060037 001122 ADD RO,SBDAOR ;ADD RH11 BASE ADDRESS
(3) 023000 012737 100000 001124 MOV #ATA,$GDDAT ;WHAT REGISTER SHOULD BE
(3) 023006 013737 001126 001164 MOV SBDDAT,$TMP0 ;MOVE REGISTER CONTENTS TO '$TMP0'
(3) 023014 042737 077777 001164 BIC #ICATA,$TMP0 ;SAVE SPECIFIED BITS
(3) 023022 023737 001124 001164 CMP $GDDAT,$TMP0 ;COMPARE THE BITS
(3) 023030 001414 BEQ 66$ ;BR IF OK
(3) 023032 013737 001126 001174 MOV SBDDAT,$TMP4 ;COPY 'BAD DATA'
(3) 023040 042737 100000 001174 BIC #ATA,$TMP4 ;CLEAR THE MASKED BITS
(3) 023046 053737 001174 001124 BIS $TMP4,$GDDAT ;'OR' WITH GOOD DATA FOR TYPEOUT
(3) 023054 104010 ERROR 10 ;REPORT THE ERROR
(3) 023056 005137 001244 COM CKERR ;SET THE REGISTER COMPARE ERROR INDICATOR
(3) 023062 000240 66$: NOP
(2) 023064 005737 001244 TST CKERR ;WAS ATTN BIT FOR PORT A SET ?
(2) 023070 001402 BEQ .+6 ;BR IF IT WAS
(2) 023072 000137 024036 JMP 1$ ;BYPASS REST OF TEST IF NOT
(3) 023076 113760 001226 000010 MOVB PORTB,RPCS2(RO) ;SELECT PORT B
(3) 023104 013737 001226 001234 MOV PORTB,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
(3) 023112 005037 001244 CLR CKERR ;CLEAR THE 'CHECK ERROR' INDICATOR
(3) 023116 016037 000012 001126 MOV RPDS1(RO),SBDDAT ;GET CONTENTS OF RPDS1
(3) 023124 012737 000012 001122 MOV #RPDS1,SBDAOR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
(3) 023132 060037 001122 ADD RO,SBDAOR ;ADD RH11 BASE ADDRESS
(3) 023136 012737 100000 001124 MOV #ATA,$GDDAT ;WHAT REGISTER SHOULD BE
(3) 023144 013737 001126 001164 MOV SBDDAT,$TMP0 ;MOVE REGISTER CONTENTS TO '$TMP0'
(3) 023152 042737 077777 001164 BIC #ICATA,$TMP0 ;SAVE SPECIFIED BITS
(3) 023160 023737 001124 001164 CMP $GDDAT,$TMP0 ;COMPARE THE BITS
(3) 023166 001414 BEQ 68$ ;BR IF OK
(3) 023170 013737 001126 001174 MOV SBDDAT,$TMP4 ;COPY 'BAD DATA'
(3) 023176 042737 100000 001174 BIC #ATA,$TMP4 ;CLEAR THE MASKED BITS
(3) 023204 053737 001174 001124 BIS $TMP4,$GDDAT ;'OR' WITH GOOD DATA FOR TYPEOUT
(3) 023212 104010 ERROR 10 ;REPORT THE ERROR
(3) 023214 005137 001244 COM CKERR ;SET THE REGISTER COMPARE ERROR INDICATOR
(3) 023220 000240 68$: NOP
(2) 023222 005737 001244 TST CKERR ;WAS ATTN BIT FOR PORT B SET ?
(2) 023226 001402 BEQ .+6 ;BR IF IT WAS

```

F07

```

(2) 023230 000137 024036          JMP      IS          ;BYPASS REST OF TEST IF NOT
(1)
(2)
(2)
(2)
(2) 023234 005037 001250          CLR      RELERR      ;CLEAR THE 'RELEASE ERROR' INDICATOR
(2) 023240 012737 000012 001122      MOV      #RPDS1,$BDAOR ;FORM THE ADDRESS OF RPDS1 FOR TYPEOUT
(2) 023246 060037 001122          ADD      RD,$BDAOR    ;ADD THE I/O BASE ADDRESS
(2) 023252 012737 111700 001124      MOV      #111700,$GDDAT ;COMPARISON CONSTANT
(2) 023260 113760 001224 000010      MOV      PORTA,RPCS2(RO) ;SELECT PORT A.
(2) 023266 016037 000012 001170      MOV      RPDS1(RO),$TMP2 ;GET THE DRIVE STATUS REGISTER FROM PORT A.
(2) 023274 013737 001170 001164      MOV      $TMP2,$TMP0    ;COPY IT INTO '$TMP0'
(2) 023302 042737 100100 001164      BIC      #ATA!VV,$TMP0  ;CLEAR PORT DEPENDENT BITS FROM THE COPY
(2) 023310 113760 001226 000010      MOV      PORTB,RPCS2(RO) ;SELECT PORT B.
(2) 023316 016037 000012 001172      MOV      RPDS1(RO),$TMP3 ;GET THE DRIVE STATUS REGISTER FROM PORT B.
(2) 023324 013737 001172 001166      MOV      $TMP3,$TMP1    ;COPY IT INTO '$TMP1'
(2) 023332 042737 100100 001166      BIC      #ATA!VV,$TMP1  ;CLEAR PORT DEPENDENT BITS FROM THE COPY
(2) 023340 023737 001164 001166      CMP      $TMP0,$TMP1    ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
(2) 023346 001006          BNE      70$          ;BR IF NOT
(2) 023350 005737 001164          TST      $TMP0        ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
(2) 023354 001045          BNE      72$          ;BR IF NOT
(2) 023356 104046          ERROR    46          ;REPORT DRIVE NOT IN NEUTRAL OR NOT SEIZED
(2) 023360 000137 023544          JMP      74$          ;BYPASS THE REST OF THE CHECKS
(2) 023364 013737 001170 001126 70$:  MOV      $TMP2,$BDDAT  ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
(2) 023372 013737 001226 001234      MOV      PORTB,PTNBR   ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
(2) 023400 113760 001226 000010      MOV      PORTB,RPCS2(RO) ;SELECT PORT B.
(2) 023406 005737 001164          TST      $TMP0        ;SEE IF STATUS EQ 0 FROM PORT A.
(2) 023412 001414          BEQ      71$          ;BR IF ZERO
(2) 023414 013737 001224 001234      MOV      PORTA,PTNBR   ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
(2) 023422 013737 001172 001126      MOV      $TMP3,$BDDAT  ;'BAD DATA' FOR ERROR TYPE OUT
(2) 023430 113760 001224 000010      MOV      PORTA,RPCS2(RO) ;SELECT PORT A.
(2) 023436 005737 001166          TST      $TMP1        ;SEE IF STATUS EQ ZERO FROM PORT B.
(2) 023442 001012          BNE      72$          ;BR IF NOT
(2) 023444 012737 177777 001250 71$:  MOV      #-1,RELERR    ;SET 'RELEASE ERROR' INDICATOR
(2) 023452 012760 000011 000000      MOV      #11,RPCS1(RO) ;CLEAR THE DRIVE
(2) 023460 012760 000013 000000      MOV      #13,RPCS1(RO) ;RELEASE THE DRIVE
(2) 023466 104026          ERROR    26          ;TYPE ERROR MESSAGE 26
(2) 023470 013737 001170 001126 72$:  MOV      $TMP2,$BDDAT  ;LOOK FOR BIT FAILURES WHEN RPDS1 READ
(2) 023476 013737 001224 001234      MOV      PORTA,PTNBR   ;CHANGE PORT NUMBER
(2) 023504 023737 001124 001170      CMP      $GDDAT,$TMP2  ;ALL BITS OK ?
(2) 023512 001401          BEQ      73$          ;BR IF OK FROM PORT A.
(2) 023514 104007          ERROR    7          ;REPORT ERROR
(2) 023516 013737 001172 001126 73$:  MOV      $TMP3,$BDDAT  ;CHECK RPDS1 FOR BIT FAILURES - FROM PORT B.
(2) 023524 013737 001226 001234      MOV      PORTB,PTNBR   ;CHANGE PORT NUMBER
(2) 023532 023737 001124 001172      CMP      $GDDAT,$TMP3  ;SEE IF READ OK FROM PORT B.
(2) 023540 001401          BEQ      74$          ;BR IF OK
(2) 023542 104007          ERROR    7          ;REPORT ERROR
(2) 023544 000240          NOP
(2) 023546 005737 001250          TST      RELERR      ;WAS DRIVE IN NEUTRAL ?
(1) 023552 001402          BEQ      .+6          ;BR IF IT WAS
(1) 023554 000137 024036          JMP      IS          ;BYPASS RESET OF TEST
(2)
(2)
(2)
(2)
(2) 023560 012760 000040 000010      MOV      #CLR,RPCS2(RO) ;ISSLE A MASSBUS INIT

```

(N)				
(N)	023566	113760	001224	000010
(N)	023574	013737	001224	001234
(N)	023602	005037	001244	
(N)	023606	016037	000012	001126
(N)	023614	012737	000012	001122
(N)	023622	060037	001122	
(N)	023626	012737	100000	001124
(N)	023634	013737	001126	001164
(N)	023642	042737	077777	001164
(N)	023650	023737	001124	001164
(N)	023656	001414		
(N)	023660	013737	001126	001174
(N)	023666	042737	100000	001174
(N)	023674	053737	001174	001124
(N)	023702	104051		
(N)	023704	005137	001244	
(N)	023710	000240		
(N)	023712	113760	001226	000010
(N)	023720	013737	001226	001234
(N)	023726	005037	001244	
(N)	023732	016037	000012	001126
(N)	023740	012737	000012	001122
(N)	023746	060037	001122	
(N)	023752	012737	100000	001124
(N)	023760	013737	001126	001164
(N)	023766	042737	077777	001164
(N)	023774	023737	001124	001164
(N)	024002	001414		
(N)	024004	013737	001126	001174
(N)	024012	042737	100000	001174
(N)	024020	053737	001174	001124
(N)	024026	104051		
(N)	024030	005137	001244	
(N)	024034	000240		
(N)	024036	000004		
(N)	024040			
(N)	024040	005737	001274	
(N)	024044	001406		

```

:*****
:CHECK THE ATTENTION BITS OF BOTH PORTS
:*****
MOV B    PORTA,RPCS2(RO) ;SELECT PORT A
MOV     PORTA,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
CLR     CKERR ;CLEAR THE 'CHECK ERROR' INDICATOR
MOV     RPDS1(RO), $BDDAT ;GET CONTENTS OF RPDS1
MOV     #RPDS1,$BDAOR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
ADD     RO,$BDAOR ;ADD RHI1 BASE ADDRESS
MOV     #ATA,$GDDAT ;WHAT REGISTER SHOULD BE
MOV     $BDDAT,$TMPD ;MOVE REGISTER CONTENTS TO '$TMPD'
BIC     #CATA,$TMPD ;SAVE SPECIFIED BITS
CMP     $GDDAT,$TMPD ;COMPARE THE BITS
BEQ     75$ ;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   51 ;TYPE MESSAGE 51
COM     CKERR ;SET THE REGISTER COMPARE ERROR INDICATOR

75$:    NOP
MOV B    PORTB,RPCS2(RO) ;SELECT PORT B
MOV     PORTB,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
CLR     CKERR ;CLEAR THE 'CHECK ERROR' INDICATOR
MOV     RPDS1(RO), $BDDAT ;GET CONTENTS OF RPDS1
MOV     #RPDS1,$BDAOR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
ADD     RO,$BDAOR ;ADD RHI1 BASE ADDRESS
MOV     #ATA,$GDDAT ;WHAT REGISTER SHOULD BE
MOV     $BDDAT,$TMPD ;MOVE REGISTER CONTENTS TO '$TMPD'
BIC     #CATA,$TMPD ;SAVE SPECIFIED BITS
CMP     $GDDAT,$TMPD ;COMPARE THE BITS
BEQ     77$ ;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   51 ;TYPE MESSAGE 51
COM     CKERR ;SET THE REGISTER COMPARE ERROR INDICATOR

77$:    NOP
1$:     SCOPE ;LOOP ?

```

```

:*****
:TEST 17 TEST SEIZE BY RPDS1 READ THROUGH PORT 'A'
:
:VERIFY THAT READING THE CONTROL REGISTER (RPDS1) SEIZES THE DRIVE.
:
:A. READ THE CONTROL REGISTER (RPDS1) 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.
:*****
†ST17: TST     KYBCTL ;PERFORMING ONLY SINGLE TESTS ?
       BEQ     2$ ;BR IF NOT

```

H07

```

(3) 024046 100002          BPL      1$          ;BR IF JUST ENTERED TEST
(3) 024050 000137 002622    JMP      EXEC        ;RETURN & GET NEXT TEST NUMBER
(3) 024054 012737 177777 001274 1$:  MOV     #-1,KYBCTL   ;SET SINGLE TEST INDICATOR
(3) 024062 112737 000017 001102 2$:  MOVVB  #17,$TSTNM   ;TEST NUMBER
(3) 024070 012737 024112 0C1106    MOV     #TEST17,$LPADR ;LOAD LOOP ON TEST ADDRESS
(3) 024076 012737 024112 001110    MOV     #TEST17,$LPERR ;LOAD LOOP ON ERROR ADDRESS
(3) 024104 012737 007640 001176    MOV     #4000,$TIMES   ;DO 4000. ITERATIONS
8530 024112 012706 001100    TEST17: MOV    #STACK,SP ;LOAD THE STACK POINTER
8541
      ;CLEAR ATTENTION BITS FOR BOTH PORTS
(3) 024116 113760 001224 000010    MOVVB  PORTA,RPCS2(RO) ;SELECT PORT #A
(3) 024124 005060 000012          CLR     RPS1(RO)      ;SEIZE THE DRIVE
(3) 024130 012760 000011 000000    MOV     #11,RPCS1(RO) ;ISSUE DRIVE CLEAR
(3) 024136 012760 000013 000000    MOV     #13,RPCS1(RO) ;RELEASE THE DRIVE
(3) 024144 113760 001226 000010    MOVVB  PORTB,RPCS2(RO) ;SELECT PORT #B
(3) 024152 005060 000012          CLR     RPS1(RO)      ;SEIZE THE DRIVE THROUGH PORT 'B'
(3) 024156 012760 000011 000000    MOV     #11,RPCS1(RO) ;ISSUE DRIVE CLEAR
(3) 024164 012760 000013 000000    MOV     #13,RPCS1(RO) ;RELEASE THE DRIVE
      ;:*****
      ;SEIZE THE DRIVE THROUGH PORT A
(3) 024172 113760 001224 000010    MOVVB  PORTA,RPCS2(RO) ;SELECT PORT A
(3) 024200 013737 001224 001236    MOV     PORTA,SEIZPT ;STORE SEIZING PORT'S ADDRESS
(3) 024206 005760 000000          TST     RPCS1(RO)    ;READ RHCS1
(3) 024212 113760 001226 000010    MOVVB  PORTB,RPCS2(RO) ;SELECT PORT B
(3) 024220 013737 001226 001234    MOV     PORTB,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
(3) 024226 013737 001226 001240    MOV     PORTB,OPPRT  ;'OPPOSITE' PORT ADDRESS
(3) 024234 016037 000012 001126    MOV     RPS1(RO),%BDDAT ;SEE IF DRIVE SEIZED BY PORT A
(3) 024242 010037 001122          MOV     RO,%BADDR    ;RH11 BASE ADDRESS
(3) 024246 062737 000012 001122    ADD     #RPS1,%BADDR ;GENERATE BAD REGISTER ADDRESS
(3) 024254 005037 001124          CLR     %GDDAT      ;REGISTER SHOULD BE ZERO
(3) 024260 023737 001124 001126    CMP     %GDDAT,%BDDAT ;IS THE REGISTER ZERO
(3) 024266 001403          BEQ     64$         ;BR IF IT IS
(3) 024270 104004          ERROR  4           ;REPORT THE ERROR
(3) 024272 000137 024724          JMP     1$         ;BYPASS REST OF THE SUBTEST
(3) 024276 113760 001224 000010    MOVVB  PORTA,RPCS2(RO) ;SELECT PORT A
(3) 024304 013737 001224 001234    MOV     PORTA,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
(3) 024312 016037 000012 001126    MOV     RPS1(RO),%BDDAT ;SEE IF SEIZING PORT SEES CORRECT STATUS
(3) 024320 012737 011700 001124    MOV     #MOL!PGM!OPR!DRY!VV,%GDDAT ;EXPECTED STATUS
(3) 024326 013737 001124 001166    MOV     %GDDAT,%STMP1 ;USE GOOD DATA AS A MASK
(3) 024334 005137 001166          COM     %STMP1      ;COMPLEMENT THE EXPECTED STATUS
(3) 024340 013737 001126 001164    MOV     %BDDAT,%STMP0 ;SAVE THE ACTUAL STATUS
(3) 024346 043737 001166 001164    BIC     %STMP1,%STMP0 ;CLEAR UNWANTED BITS
(3) 024354 023737 001124 001164    CMP     %GDDAT,%STMP0 ;ARE THE EXPECTED STATUS BITS SET ?
(3) 024362 0C1401          BEQ     65$         ;BR IF THEY ARE
(3) 024364 104005          ERROR  5           ;REPORT THE ERROR
(3) 024366 000240          NOP
      ;:*****
      ;RELEASE THE DRIVE FROM PORT A

```

```

(3) 024370 113760 001224 000010      MOVB   PORTA,RPCS2(RO) ;SELECT PORT A
(3) 024376 013737 001224 001234      MOV    PORTA,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
(2) 024404 012760 000013 000000      MOV    #13,RPCS1(RO) ;ISSUE RELEASE THROUGH PORT A
(3)                                     ;VERIFY THAT THE DRIVE IS IN NEUTRAL
(3) 024412 005037 001250                                     CLR    RELERR ;CLEAR THE 'RELEASE ERROR' INDICATOR
(3) 024416 012737 000012 001122      MOV    #RPDS1,$BDDADR ;FORM THE ADDRESS OF RPDS1 FOR TYPEOUT
(3) 024424 060037 001122                                     ADD    RO,$BDDADR ;ADD THE I/O BASE ADDRESS
(3) 024430 012737 011700 001124      MOV    #MOL:PGM:DPR:DRY:VV,$GDDAT ;COMPARISON CONSTANT
(3) 024436 113760 001224 000010      MOVB   PORTA,RPCS2(RO) ;SELECT PORT A.
(3) 024444 016037 000012 001170      MOV    RPDS1(RO),STMP2 ;GET THE DRIVE STATUS REGISTER FROM PORT A.
(3) 024452 013737 001170 001164      MOV    STMP2,STMP0 ;COPY IT INTO 'STMP0'
(3) 024460 042737 100100 001164      BIC    #ATA:VV,STMP0 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
(3) 024466 113760 001226 000010      MOVB   PORTB,RPCS2(RO) ;SELECT PORT B.
(3) 024474 016037 000012 001172      MOV    RPDS1(RO),STMP3 ;GET THE DRIVE STATUS REGISTER FROM PORT B.
(3) 024502 013737 001172 001166      MOV    STMP3,STMP1 ;COPY IT INTO 'STMP1'
(3) 024510 042737 100100 001166      BIC    #ATA:VV,STMP1 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
(3) 024516 023737 001164 001166      CMP    STMP0,STMP1 ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
(3) 024524 001006                                     BNE    66$ ;BR IF NOT
(3) 024526 005737 001164                                     TST   STMP0 ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
(3) 024532 001045                                     BNE    68$ ;BR IF NOT
(3) 024534 104046      ERROR 46 ;REPORT DRIVE NOT IN NEUTRAL OR NOT SEIZED
(3) 024536 000137 024722                                     JMP   70$ ;BYPASS THE REST OF THE CHECKS
(3) 024542 013737 001170 001126 66$:   MOV    STMP2,$BDDAT ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
(3) 024550 013737 001226 001234      MOV    PORTB,PTNBR ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
(3) 024556 113760 001226 000010      MOVB   PORTB,RPCS2(RO) ;SELECT PORT B.
(3) 024564 005737 001164      TST   STMP0 ;SEE IF STATUS EQ 0 FROM PORT A.
(3) 024570 001414      BEQ   67$ ;BR IF ZERO
(3) 024572 013737 001224 001234      MOV    PORTA,PTNBR ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
(3) 024600 013737 001172 001126      MOV    STMP3,$BDDAT ;'BAD DATA' FOR ERROR TYPE OUT
(3) 024606 113760 001224 000010      MOVB   PORTA,RPCS2(RO) ;SELECT PORT A.
(3) 024614 005737 001166      TST   STMP1 ;SEE IF STATUS EQ ZERO FROM PORT B.
(3) 024620 001012      SNE   68$ ;BR IF NOT
(3) 024622 012737 177777 001250 67$:   MOV    #-1,RELERR ;SET 'RELEASE ERROR' INDICATOR
(3) 024630 012760 000011 000000      MOV    #11,RPCS1(RO) ;CLEAR THE DRIVE
(3) 024636 012760 000013 000000      MOV    #13,RPCS1(RO) ;RELEASE THE DRIVE
(3) 024644 104026      ERROR 26 ;TYPE ERROR MESSAGE 26
(3) 024646 013737 001170 001126 68$:   MOV    STMP2,$BDDAT ;LOOK FOR BIT FAILURES WHEN RPDS1 REAC
(3) 024654 013737 001224 001234      MOV    PORTA,PTNBR ;CHANGE PORT NUMBER
(3) 024662 023737 001124 001170      CMP    $GDDAT,STMP2 ;ALL BITS OK ?
(3) 024670 001401      BEQ   69$ ;BR IF OK FROM PORT A.
(3) 024672 104007      ERROR 7 ;REPORT ERROR
(3) 024674 013737 001172 001126 69$:   MOV    STMP3,$BDDAT ;CHECK RPDS1 FOR BIT FAILURES - FROM PORT B.
(3) 024702 013737 001226 001234      MOV    PORTB,PTNBR ;CHANGE PORT NUMBER
(3) 024710 023737 001124 001172      CMP    $GDDAT,STMP3 ;SEE IF READ OK FROM PORT B.
(3) 024716 001401      BEQ   70$ ;BR IF OK
(3) 024720 104007      ERROR 7 ;REPORT ERROR
(3) 024722 000240      NOP
(1) 024724 000004      IS:   SCOPE ;LOOP ?

```

8553
8554
(3)
(4)
4,
4,

```

*****
;TEST 20 TEST SEIZE BY RPCS1 READ THROUGH PORT 'B'
;
;VERIFY THAT READING THE CONTROL REGISTER (RPCS1) SEIZES THE DRIVE.
;

```

```

(4) * A. READ THE CONTROL REGISTER (RPCS1) THROUGH PORT 'B'; VERIFY THAT
(4) * THE DRIVE IS SEIZED.
(4) *
(4) * B. ISSUE A RELEASE COMMAND THROUGH PORT 'B'; VERIFY THAT THE DRIVE
(4) * RETURNED TO NEUTRAL AND THAT NEITHER ATTENTION BIT IS SET.
(4) *
(3) *****
(2)

```

```

(2) 024726 005737 001274
(3) 024726 001406
(3) 024732 100002
(3) 024734 000137 002622
(3) 024736 012737 177777 001274 1$: MOV #1,KYBCTL ;SET SINGLE TEST INDICATOR
(3) 024742 012737 000020 001102 2$: MOVB #20,$STNM ;TEST NUMBER
(3) 024750 012737 025000 001106 MOV #TEST20,$LPADR ;LOAD LOOP ON TEST ADDRESS
(3) 024756 012737 025000 001110 MOV #TEST20,$LPERR ;LOAD LOOP ON ERROR ADDRESS
(1) 024772 012737 007640 001176 MOV #4000,$TIMES ;DO 4000. ITERATIONS
8555 025000 012706 001100 TEST20: MOV #STACK,$P ;LOAD THE STACK POINTER
8556
(2) ;CLEAR ATTENTION BITS FOR BOTH PORTS
(2) MOVB PORTA,RPCS2(RO) ;SELECT PORT #A
(2) 025004 113760 001224 000010 CLR RPDS1(RO) ;SEIZE THE DRIVE
(2) 025012 005060 000012 MOV #11,RPCS1(RO) ;ISSUE DRIVE CLEAR
(2) 025016 012760 000011 000000 MOV #13,RPCS1(RO) ;RELEASE THE DRIVE
(2) 025024 012760 000013 000000 MOVB PORTB,RPCS2(RO) ;SELECT PORT #B
(2) 025032 113760 001226 000010 CLR RPDS1(RO) ;SEIZE THE DRIVE THROUGH PORT 'B'
(2) 025040 005060 000012 MOV #11,RPCS1(RO) ;ISSUE DRIVE CLEAR
(2) 025044 012760 000011 000000 MOV #13,RPCS1(RO) ;RELEASE THE DRIVE
(2) 025052 012760 000013 000000
(1)
(2) ;*****
(2)
(2) ;SEIZE THE DRIVE THROUGH PORT B
(2) MOVB PORTB,RPCS2(RO) ;SELECT PORT B
(2) 025060 113760 001226 000010 MOV PORTB,SEIZPT ;STORE SEIZING PORT'S ADDRESS
(2) 025066 013737 001226 001236 TST RPCS1(RO) ;READ RPCS1
(2) 025074 005760 000000 MOVB PORTA,RPCS2(RO) ;SELECT PORT A
(3) 025100 113760 001224 000010 MOV PORTA,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
(3) 025106 013737 001224 001234 MOV PORTA,OPPR ;'OPPOSITE' PORT ADDRESS
(2) 025114 013737 001224 001240 MOV RPDS1(RO),$BDDAT ;SEE IF DRIVE SEIZED BY PORT B
(2) 025122 016037 000012 001126 MOV RO,$BDAOR ;R#11 BASE ADDRESS
(2) 025130 010037 001122 ADD #RPDS1,$BDAOR ;GENERATE BAD REGISTER ADDRESS
(2) 025134 062737 000012 001122 CLR $GDDAT ;REGISTER SHOULD BE ZERO
(2) 025142 005037 001124 CMP $GDDAT,$BDDAT ;IS THE REGISTER ZERO
(2) 025146 023737 001124 001126 BEQ 64$ ;BR IF IT IS
(2) 025154 001403 ERROR 4 ;REPORT THE ERROR
(2) 025156 104004 JMP 1$ ;BYPASS REST OF THE SUBTEST
(2) 025160 000137 025612 64$:
(2) 025164 MOVB PORTB,RPCS2(RO) ;SELECT PORT B
(3) 025164 113760 001226 000010 MOV PORTB,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
(3) 025172 013737 001226 001234 MOV RPDS1(RO),$BDDAT ;SEE IF SEIZING PORT SEES CORRECT STATUS
(2) 025200 016037 000012 001126 MOV #MOL:PGM:OPR:DRY:VV,$GDDAT ;EXPECTED STATUS
(2) 025206 012737 011700 001124 MOV $GDDAT,$TMP1 ;USE GOOD DATA AS A MASK
(2) 025214 013737 001124 001166 COM $TMP1 ;COMPLEMENT THE EXPECTED STATUS
(2) 025222 005137 001166 MOV $BDDAT,$TMP0 ;SAVE THE ACTUAL STATUS
(2) 025226 013737 001126 001164

```


K07

```

(2) 025234 043737 001166 001164      BIC    $TMP1,$TMP0      ;CLEAR UNWANTED BITS
(2) 025242 023737 001124 001164      CMP    $GDDAT,$TMP0    ;ARE THE EXPECTED STATUS BITS SET
(2) 025250 001401                      BEQ    65$              ;BR IF THEY ARE
(2) 025252 104005                      ERROR 5                 ;REPORT THE ERROR
(2) 025254 000240                      65$: NOP

;*****

;RELEASE THE DRIVE FROM PORT B
(3) 025256 113760 001226 000010      MOV    PORTB,RPCS2(RO) ;SELECT PORT B
(3) 025264 013737 001226 001234      MOV    PORTB,PTNBR     ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
(3) 025272 012760 000013 000000      MOV    #13,RPCS1(RO)  ;ISSUE RELEASE THROUGH PORT B

;VERIFY THAT THE DRIVE IS IN NEUTRAL
(3) 025300 005037 001250                      CLR    RELERR          ;CLEAR THE 'RELEASE ERROR' INDICATOR
(3) 025304 012737 000012 001122      MOV    #RPDS1,$BDAOR   ;FORM THE ADDRESS OF RPDS1 FOR TYPEOUT
(3) 025312 060037 001122                      ADD    RO,$BDAOR       ;ADD THE I/O BASE ADDRESS
(3) 025316 012737 011700 001124      MOV    #MOL:PGM:DPR:DRY!VV,$GDDAT ;COMPARISON CONSTANT
(3) 025324 113760 001224 000010      MOV    PORTA,RPCS2(RO) ;SELECT PORT A.
(3) 025332 016037 000012 001170      MOV    RPDS1(RO),$TMP2 ;GET THE DRIVE STATUS REGISTER FROM PORT A.
(3) 025340 013737 001170 001164      MOV    $TMP2,$TMP0     ;COPY IT INTO '$TMP0'
(3) 025346 042737 100100 001164      BIC    #ATA!VV,$TMP0   ;CLEAR PORT DEPENDENT BITS FROM THE COPY
(3) 025354 113760 001226 000010      MOV    PORTB,RPCS2(RO) ;SELECT PORT B.
(3) 025362 016037 000012 001172      MOV    RPDS1(RO),$TMP3 ;GET THE DRIVE STATUS REGISTER FROM PORT B.
(3) 025370 013737 001172 001166      MOV    $TMP3,$TMP1    ;COPY IT INTO '$TMP1'
(3) 025376 042737 100100 001166      BIC    #ATA!VV,$TMP1   ;CLEAR PORT DEPENDENT BITS FROM THE COPY
(3) 025404 023737 001164 001166      CMP    $TMP0,$TMP1    ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
(3) 025412 001006                      BNE    66$             ;BR IF NOT
(3) 025414 005737 001164                      TST    $TMP0           ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
(3) 025420 001045                      BNE    68$             ;BR IF NOT
(3) 025422 104046                      ERROR 46                ;REPORT DRIVE NOT IN NEUTRAL OR NOT SEIZED
(3) 025424 000137 025610                      JMP    70$             ;BYPASS THE REST OF THE CHECKS
(3) 025430 013737 001170 001126 66$: MOV    $TMP2,$BDDAT    ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
(3) 025436 013737 001226 001234      MOV    PORTB,PTNBR     ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
(3) 025444 113760 001226 000010      MOV    PORTB,RPCS2(RO) ;SELECT PORT B.
(3) 025452 005737 001164                      TST    $TMP0           ;SEE IF STATUS EQ 0 FROM PORT A.
(3) 025456 001414                      BEQ    67$             ;BR IF ZERO
(3) 025460 013737 001224 001234      MOV    PORTA,PTNBR     ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
(3) 025466 013737 001172 001126      MOV    $TMP3,$BDDAT    ;'BAD DATA' FOR ERROR TYPE OUT
(3) 025474 113760 001224 000010      MOV    PORTA,RPCS2(RO) ;SELECT PORT A.
(3) 025502 005737 001166                      TST    $TMP1           ;SEE IF STATUS EQ ZERO FROM PORT B.
(3) 025506 001012                      BNE    68$             ;BR IF NOT
(3) 025510 012737 177777 001250 67$: MOV    #-1,RELERR     ;SET 'RELEASE ERROR' INDICATOR
(3) 025516 012760 000011 000000      MOV    #11,RPCS1(RO)  ;CLEAR THE DRIVE
(3) 025524 012760 000013 000000      MOV    #13,RPCS1(RO)  ;RELEASE THE DRIVE
(3) 025532 104026                      ERROR 26                ;TYPE ERROR MESSAGE 26
(3) 025534 013737 001170 001126 68$: MOV    $TMP2,$BDDAT    ;LOOK FOR BIT FAILURES WHEN RPDS1 READ
(3) 025542 013737 001224 001234      MOV    PORTA,PTNBR     ;CHANGE PORT NUMBER
(3) 025550 023737 001124 001170      CMP    $GDDAT,$TMP2   ;ALL BITS OK ?
(3) 025556 001401                      BEQ    69$             ;BR IF OK FROM PORT A.
(3) 025560 104007                      ERROR 7                 ;REPORT ERROR
(3) 025562 013737 001172 001126 69$: MOV    $TMP3,$BDDAT    ;CHECK RPDS1 FOR BIT FAILURES - FROM PORT B.
(3) 025570 013737 001226 001234      MOV    PORTB,PTNBR     ;CHANGE PORT NUMBER
(3) 025576 023737 001124 001172      CMP    $GDDAT,$TMP3   ;SEE IF READ OK FROM PORT B.
    
```

(3) 025604 001401
 (3) 025606 104007
 (3) 025610 000240
 (1) 025612 000004
 8578
 8610
 (2)
 (2)
 (2) 025614 005737 001274
 (3) 025614 001406
 (3) 025622 100002
 (3) 025624 000137 002622
 (3) 025630 012737 177777 001274
 (3) 025636 112737 000021 001102
 (3) 025644 012737 025666 001106
 (3) 025652 012737 025666 001110
 (1) 025660 012737 007640 001176
 (2) 025666 012706 001100
 (2)
 (2) 025672 113760 001224 000010
 (2) 025700 005060 000012
 (2) 025704 012760 000011 000000
 (2) 025712 012760 000013 000000
 (2) 025720 113760 001226 000010
 (2) 025726 005060 000012
 (2) 025732 012760 000011 000000
 (2) 025740 012760 000013 000000
 (1)
 (2)
 (2)
 (2) 025746 113760 001226 000010
 (2) 025754 013737 001226 001236
 (2) 025762 005060 000012
 (2) 025766 013737 001224 001240

```

BEQ 705 ;BR IF OK
ERROR 7 ;REPORT ERROR
705: NOP
15: 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.
*
*****
†ST21:
TST KYBCTL ;PERFORMING ONLY SINGLE TESTS ?
BEQ 25 ;BR IF NOT
BPL 15 ;BR IF JUST ENTERED TEST
JMP EXEC ;RETURN & GET NEXT TEST NUMBER
15: MOV #-1,KYBCTL ;SET SINGLE TEST INDICATOR
25: MOVB #21,$TSTNM ;TEST NUMBER
MOV #TEST21,$LPADR ;LOAD LOOP ON TEST ADDRESS
MOV #TEST21,$LPERR ;LOAD LOOP ON ERROR ADDRESS
MOV #4000,$TIMES ;DO 4000. ITERATIONS
TEST21: MOV #STACK,$SP ;LOAD THE STACK POINTER

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

*****
;SEIZE THE DRIVE THROUGH PORT B
MOVB PORTB,RPCS2(RO) ;SELECT PORT B
MOV PORTB,SEIZPT ;STORE SEIZING PORT'S ADDRESS
CLR RPDS1(RO) ;WRITE RPDS1
MOV PORTA,OPPRT ;'OPPOSITE' PORT ADDRESS

```

M07

CZRJECO. DL CTRLR LGC MACY11 30A(1052) 28-DEC-77 10:17 PAGE 66-56
CZRJEC.P11 21-DEC-77 14:19 T21 TEST 'PORT REQUEST' FROM PORT 'A'

SEG 0090

```

(2) 025774 113760 001224 000010      MOV B   PORTA,RPCS2(RO) ;SELECT PORT A
(2) 026002 013737 001224 001234      MOV     PORTA,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
(1)
(2)                                     ;*****
(1)                                     ;SET PORT REQUEST
(1)
(1) 026010 005060 000012                CLR     RPDS1(RO) ;SET PORT REQUEST FOR PORT A
(1)
(2)                                     ;*****
(1)                                     ;RELEASE THROUGH PORT B. DRIVE SHOULD SWITCH TO PORT A.
(1)
(2)
(2)                                     ;RELEASE THE DRIVE FROM PORT B
(2)
(3) 026014 113760 001226 000010      MOV B   PORTB,RPCS2(RO) ;SELECT PORT B
(3) 026022 013737 001226 001234      MOV     PORTB,PTNBR ;MOVE PORT ADDRESS TO LOCAT:JN FOR TYPEOUT
(2) 026030 012760 000013 000000      MOV     #13,RPDS1(RO) ;ISSUE RELEASE THROUGH PORT B
(3)
(3)                                     ;VERIFY THAT DRIVE IS SEIZED BY PORT A WHEN RELEASED BY PORT B
(3)
(3) 026036 005037 001250                CLR     RELERR ;CLEAR 'RELEASE ERROR' INDICATOR
(3) 026042 012737 111700 001124      MOV     #ATA!MOL!PGM!DPR!DRY!VV,$GDDAT ;COMPARISON CONSTANT
(3) 026050 012737 000012 001122      MOV     #RPDS1,$BDDADR ;REGISTER ADDRESS INCREMENT
(3) 026056 060037 001122                ADD     RO,$BDDADR ;REGISTER BASE ADDRESS FOR TYPEOUT
(4) 026062 113760 001224 000010      MOV B   PORTA,RPCS2(RO) ;SELECT PORT A
(4) 026070 013737 001224 001234      MOV     PORTA,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
(3) 026076 016037 000012 001164      MOV     RPDS1(RO),$TMP0 ;READ STATUS REGISTER FROM PORT A
(4) 026104 113760 001226 000010      MOV B   PORTB,RPCS2(RO) ;SELECT PORT B
(4) 026112 013737 001226 001234      MOV     PORTB,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT ;
(3) 026120 016037 000012 001126      MOV     RPDS1(RO),$BDDAT ;DRIVE STATUS FROM PORT B
(3) 026126 001404                BEQ     66$ ;BR IF STATUS FROM PORT B ZERO
(3) 026130 005737 001164                TST    $TMP0 ;IS STATUS FROM PORT A ZERO ?
(3) 026134 001401                BEQ     66$ ;BR IF ZERO
(3) 026136 104031                ERROR   31 ;REPORT DRIVE IN NEUTRAL
(3) 026140 013737 001164 001126 66$: MOV     $TMP0,$BDDAT ;CHECK STATUS FROM PORT A
(3) 026146 013737 001224 001234      MOV     PORTA,PTNBR ;CHANGE PORT ADDRESS FOR TYPEOUT
(3) 026154 023737 001124 001126      CMP     $GDDAT,$BDDAT ;COMPARE WITH CONSTANT
(3) 026162 001401                BEQ     67$ ;BR IF OK
(3) 026164 104027                ERROR   27 ;REPORT REGISTER ERROR
(3) 026166 000240                NOP
(2) 026170 113760 001226 000010 67$: MOV B   PORTB,RPCS2(RO) ;SELECT PORT B
(2) 026176 013737 001226 001234      MOV     PORTB,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
(2) 026204 005037 001244                CLR     CKERR ;CLEAR THE 'CHECK ERROR' INDICATOR
(2) 026210 016037 000012 001126      MOV     RPDS1(RO),$BDDAT ;GET CONTENTS OF RPDS1
(2) 026216 012737 000012 001122      MOV     #RPDS1,$BDDADR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
(2) 026224 060037 001122                ADD     RO,$BDDADR ;ADD RH11 BASE ADDRESS
(2) 026230 005037 001124                CLR     $GDDAT ;WHAT REGISTER SHOULD BE
(2) 026234 013737 001126 001164      MOV     $BDDAT,$TMP0 ;MOVE REGISTER CONTENTS TO '$TMP0'
(2) 026242 042737 077777 001164      BIC     #!CATA,$TMP0 ;SAVE SPECIFIED BITS
(2) 026250 023737 001124 001164      CMP     $GDDAT,$TMP0 ;COMPARE THE BITS
(2) 026256 001414                BEQ     68$ ;BR IF OK
(2) 026260 013737 001126 001174      MOV     $BDDAT,$TMP4 ;COPY 'BAD DATA'
(2) 026266 042737 100000 001174      BIC     #ATA,$TMP4 ;CLEAR THE MASKED BITS
(2) 026274 053737 001174 001124      BIS     $TMP4,$GDDAT ;'OR' WITH GOOD DATA FOR TYPEOUT
(2) 026302 104016                ERROR   16 ;TYPE MESSAGE 16
(2) 026304 005137 001244                COM     CKERR ;SET THE REGISTER COMPARE ERROR INDICATOR

```

```

(2) 026310 000240      68$:  NOP
(2) 026312 113760      MOV  PORTA,RPCS2(RO) ;SELECT PORT A
(2) 026320 013737      MOV  PORTA,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
(2) 026326 005037      CLR  CKERR ;CLEAR THE 'CHECK ERROR' INDICATOR
(2) 026332 016037      MOV  RPDS1(RO),SDDAT ;GET CONTENTS OF RPDS1
(2) 026340 012737      MOV  #RPDS1,SDDADR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
(2) 026346 060037      ADD  RO,SDDADR ;ADD RHI1 BASE ADDRESS
(2) 026352 012737      MOV  #ATA,SDDAT ;WHAT REGISTER SHOULD BE
(2) 026360 013737      MOV  SDDAT,$TMP0 ;MOVE REGISTER CONTENTS TO '$TMP0'
(2) 026366 042737      BIC  #ICATA,$TMP0 ;SAVE SPECIFIED BITS
(2) 026374 023737      CMP  $SDDAT,$TMP0 ;COMPARE THE BITS
(2) 026402 001414      BEQ  70$ ;BR IF OK
(2) 026404 013737      MOV  SDDAT,$TMP4 ;COPY 'BAD DATA'
(2) 026412 042737      BIC  #ATA,$TMP4 ;CLEAR THE MASKED BITS
(2) 026420 053737      BIS  $TMP4,$SDDAT ;'OR' WITH GOOD DATA FOR TYPEOUT
(2) 026426 104016      ERROR 16 ;TYPE MESSAGE 16
(2) 026430 005137      COM  CKERR ;SET THE REGISTER COMPARE ERROR INDICATOR
(2) 026434 000240      70$: NOP

;*****
;RELEASE THE DRIVE FROM PORT A
(3) 026436 113760      MOV  PORTA,RPCS2(RO) ;SELECT PORT A
(3) 026444 013737      MOV  PORTA,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
(3) 026452 012760      MOV  #13,RPDS1(RO) ;ISSUE RELEASE THROUGH PORT A

;VERIFY THAT THE DRIVE IS IN NEUTRAL
(3) 026460 005037      CLR  RELERR ;CLEAR THE 'RELEASE ERROR' INDICATOR
(3) 026464 012737      MOV  #RPDS1,SDDADR ;FORM THE ADDRESS OF RPDS1 FOR TYPEOUT
(3) 026472 060037      ADD  RO,SDDADR ;ADD THE I/O BASE ADDRESS
(3) 026476 012737      MOV  #MOL:PGM:DPR:DRY!VV,$SDDAT ;COMPARISON CONSTANT
(3) 026504 113760      MOV  PORTA,RPCS2(RO) ;SELECT PORT A.
(3) 026512 016037      MOV  RPDS1(RO),$TMP2 ;GET THE DRIVE STATUS REGISTER FROM PORT A.
(3) 026520 013737      MOV  $TMP2,$TMP0 ;COPY IT INTO '$TMP0'
(3) 026526 042737      BIC  #ATA!VV,$TMP0 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
(3) 026534 113760      MOV  PORTB,RPCS2(RO) ;SELECT PORT B.
(3) 026542 016037      MOV  RPDS1(RO),$TMP3 ;GET THE DRIVE STATUS REGISTER FROM PORT B.
(3) 026550 013737      MOV  $TMP3,$TMP1 ;COPY IT INTO '$TMP1'
(3) 026556 042737      BIC  #ATA!VV,$TMP1 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
(3) 026564 023737      CMP  $TMP0,$TMP1 ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
(3) 026572 001006      BNE  72$ ;BR IF NOT
(3) 026574 005737      TST  $TMP0 ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
(3) 026600 001045      BNE  74$ ;BR IF NOT
(3) 026602 104046      ERROR 46 ;REPORT DRIVE NOT IN NEUTRAL OR NOT SEIZED
(3) 026604 000137      JMP  76$ ;BYPASS THE REST OF THE CHECKS
(3) 026610 013737      MOV  $TMP2,$SDDAT ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
(3) 026616 013737      MOV  PORTB,PTNBR ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
(3) 026624 113760      MOV  PORTB,RPCS2(RO) ;SELECT PORT B.
(3) 026632 005737      TST  $TMP0 ;SEE IF STATUS EQ 0 FROM PORT A.
(3) 026636 001414      BEQ  73$ ;BR IF ZERO
(3) 026640 013737      MOV  PORTA,PTNBR ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
(3) 026646 013737      MOV  $TMP3,$SDDAT ;'BAD DATA' FOR ERROR TYPE OUT
(3) 026654 113760      MOV  PORTA,RPCS2(RO) ;SELECT PORT A.
(3) 026662 005737      TST  $TMP1 ;SEE IF STATUS EQ ZERO FROM PORT B.

```

(3)	026666	001012				BNE	74\$;BR IF NOT
(3)	026670	012737	177777	001250	73\$:	MOV	#-1,RELEA		;SET 'RELEASE ERROR' INDICATOR
(3)	026676	012760	000011	000000		MOV	#11,RPCS1(RO)		;CLEAR THE DRIVE
(3)	026704	012760	000013	000000		MOV	#13,RPCS1(RO)		;RELEASE THE DRIVE
(3)	026712	104026				ERROR	26		;TYPE ERROR MESSAGE 26
(3)	026714	013737	001170	001126	74\$:	MOV	\$TMP2,\$BDDAT		;LOOK FOR BIT FAILURES WHEN RPDS1 PEAC
(3)	026722	013737	001224	001234		MOV	PORTA,PTNBR		;CHANGE PORT NUMBER
(3)	026730	023737	001124	001170		CMP	\$GDDAT,\$TMP2		;ALL BITS OK ?
(3)	026736	001401				BEQ	75\$;BR IF OK FROM PORT A.
(3)	026740	104007				ERROR	7		;REPORT ERROR
(3)	026742	013737	001172	001126	75\$:	MOV	\$TMP3,\$BDDAT		;CHECK RPDS1 FOR BIT FAILURES - FROM PORT B.
(3)	026750	013737	001226	001234		MOV	PORTB,PTNBR		;CHANGE PORT NUMBER
(3)	026756	023737	001124	001172		CMP	\$GDDAT,\$TMP3		;SEE IF READ OK FROM PORT B.
(3)	026764	001401				BEQ	76\$;BR IF OK
(3)	026766	104007				ERROR	7		;REPORT ERROR
(3)	026770	000240			76\$:	NOP			
1	026772	000004			1\$:	SCOPE			;LOOP ?

8630
8631
(3)
(4)
(4)
(4)
(4)
(4)
(4)
(4)
(4)
(4)
(4)
(4)
(4)
(4)
(4)
(4)
(3)
(2) 026774
(3) 026774
(3) 027000
(3) 027003
(3) 027004
(3) 027010
(3) 027016
(3) 027024
(3) 027032
(1) 027040
8632 027046
8633
(2)
(2)
(2) 027052
(2) 027060
(2) 027064
(2) 027072
(2) 027100
(2) 027106
(2) 027112
(2) 027120
(1)
(2)
(2)
(2)
(2) 027126
(2) 027134
(2) 027142
(2) 027146
(2) 027154
(2) 027162
(1)
2.

005737 001274
001406
100002
000137 002622
012737 177777 001274
112737 000022 001102
012737 027046 001106
012737 027046 001110
012737 007640 001176
012706 001100

113760 001224 000010
005060 000012
012760 000011 000000
012760 000013 000000
113760 001226 000010
005060 000012
012760 000011 000000
012760 000013 000000

113760 001224 000010
013737 001224 001236
005060 000012
013737 001226 001240
113760 001226 000010
013737 001226 001234

```
*****
*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.
*
*****
↑ST22:
TST KYBCTL ;PERFORMING ONLY SINGLE TESTS ?
BEG 25 ;BR IF NOT
IS ;BR IF JUST ENTERED TEST
JMP EXEC ;RETURN & GET NEXT TEST NUMBER
1$: MOV #-1,KYBCTL ;SET SINGLE TEST INDICATOR
2$: MOVB #22,$TSTNM ;TEST NUMBER
MOV #TEST22,$LPADR ;LOAD LOOP ON TEST ADDRESS
MOV #TEST22,$LPERR ;LOAD LOOP ON ERROR ADDRESS
MOV #4000,$TIMES ;DO 4000 ITERATIONS
TEST22: MOV #STACK,$SP ;LOAD THE STACK POINTER

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

*****
;SEIZE THE DRIVE THROUGH PORT A
MOVB PORTA,RPCS2(RO) ;SELECT PORT A
MOV PORTA,SEIZPT ;STORE SEIZING PORT'S ADDRESS
CLR RPDS1(RO) ;WRITE RPDS1
MOV PORTB,OPPRT ;'OPPOSITE' PORT ADDRESS
MOVB PORTB,RPCS2(RO) ;SELECT PORT B
MOV PORTB,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT

*****
```

```

(1) ;SET PORT REQUEST
(1)
(1) 027170 005060 000012 CLR RPDS1(RC) ;SET PORT REQUEST FOR PORT B
(1)
(2) ;*****
(1) ;RELEASE THROUGH PORT A. DRIVE SHOULD SWITCH TO PORT B.
(1)
(2)
(2) ;RELEASE THE DRIVE FROM PORT A
(2)
(2) 027174 113760 001224 000010 MOVB PORTA,RPCS2(R0) ;SELECT PORT A
(2) 027202 013737 001224 001234 MOV PORTA,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
(2) 027210 012760 000013 000000 MOV #13,RPDS1(R0) ;ISSUE RELEASE THROUGH PORT A
(2)
(2) ;VERIFY THAT DRIVE IS SEIZED BY PORT B WHEN RELEASED BY PORT A
(2)
(2) 027216 005037 001250 CLR RELERR ;CLEAR 'RELEASE ERROR' INDICATOR
(2) 027222 012737 111700 001124 MOV #ATA!MOL!PGM!DPR!DRY!VV,$GDDAT ;COMPARISON CONSTANT
(2) 027230 012737 000012 001122 MOV #RPDS1,$BDAOR ;REGISTER ADDRESS INCREMENT
(2) 027236 060037 001122 ADD R0,$BDAOR ;REGISTER BASE ADDRESS FOR TYPEOUT
(2) 027242 113760 001226 000010 MOVB PORTB,RPCS2(R0) ;SELECT PORT B
(2) 027250 013737 001226 001234 MOV PORTB,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
(2) 027256 016037 000012 001164 MOV RPDS1(R0),$TMP0 ;READ STATUS REGISTER FROM PORT B
(2) 027264 113760 001224 000010 MOVB PORTA,RPCS2(R0) ;SELECT PORT A
(2) 027272 013737 001234 001234 MOV PORTA,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
(2) 027300 016037 000012 001126 MOV RPDS1(R0),$BDDAT ;DRIVE STATUS FROM PORT A
(2) 027306 001404 BEQ 66$ ;BR IF STATUS FROM PORT A ZERO
(2) 027310 005737 001164 TST $TMP0 ;IS STATUS FROM PORT B ZERO?
(2) 027314 001401 BEQ 66$ ;BR IF ZERO
(2) 027316 104031 ERROR 31 ;REPORT DRIVE IN NEUTRAL
(2) 027320 013737 001164 001126 66$: MOV $TMP0,$BDDAT ;CHECK STATUS FROM PORT B
(2) 027326 013737 001226 001234 MOV PORTB,PTNBR ;CHANGE PORT ADDRESS FOR TYPEOUT
(2) 027334 023737 001124 001126 CMP $GDDAT,$BDDAT ;COMPARE WITH CONSTANT
(2) 027342 001401 BEQ 67$ ;BR IF OK
(2) 027344 104027 ERROR 27 ;REPORT REGISTER ERROR
(2) 027346 000240 67$: NOP
(2) 027350 113760 001224 000010 MOVB PORTA,RPCS2(R0) ;SELECT PORT A
(2) 027356 013737 001224 001234 MOV PORTA,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
(2) 027364 005037 001244 CLR CKERR ;CLEAR THE 'CHECK ERROR' INDICATOR
(2) 027370 016037 000012 001126 MOV RPDS1(R0),$BDDAT ;GET CONTENTS OF RPDS1
(2) 027376 012737 000012 001122 MOV #RPDS1,$BDAOR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
(2) 027404 060037 001122 ADD R0,$BDAOR ;ADD RHI1 BASE ADDRESS
(2) 027410 005037 001124 CLR $GDDAT ;WHAT REGISTER SHOULD BE
(2) 027414 013737 001126 001164 MOV $BDDAT,$TMP0 ;MOVE REGISTER CONTENTS TO '$TMP0'
(2) 027422 042737 077777 001164 BIC #!CATA,$TMP0 ;SAVE SPECIFIED BITS
(2) 027430 023737 001124 001164 CMP $GDDAT,$TMP0 ;COMPARE THE BITS
(2) 027436 001414 BEQ 68$ ;BR IF OK
(2) 027440 013737 001126 001174 MOV $BDDAT,$TMP4 ;COPY 'BAD DATA'
(2) 027446 042737 100000 001174 BIC #ATA,$TMP4 ;CLEAR THE MASKED BITS
(2) 027454 053737 001174 001124 BIS $TMP4,$GDDAT ;'OR' WITH GOOD DATA FOR TYPEOUT
(2) 027462 104016 ERROR 16 ;TYPE MESSAGE 16
(2) 027464 005137 001244 COM CKERR ;SET THE REGISTER COMPARE ERROR INDICATOR
(2) 027470 000240 68$: NOP
(2) 027472 113760 001226 000010 MOVB PORTB,RPCS2(R0) ;SELECT PORT B
(2) 027500 013737 001226 001234 MOV PORTB,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
(2) 027506 005037 001244 CLR CKERR ;CLEAR THE 'CHECK ERROR' INDICATOR

```

E08

CZRJECQ DL CTRLR LGC MACY11 30A(1052) 28-DEC-77 10:17 PAGE 67-2
CZRJEC.P11 21-DEC-77 14:19 T22 TEST PORT REQUEST FROM PORT 'B'

SEQ 0095

```
(2) 027512 016037 000012 001126 MOV RPDS1(RO), $BDDAT ; GET CONTENTS OF RPDS1
(2) 027520 012737 000012 001122 MOV #RPDS1, $BDAOR ; FORM REGISTER ADDRESS OF ERROR MESSAGE
(2) 027526 060037 001122 ADD RO, $BDAOR ; ADD RHI1 BASE ADDRESS
(2) 027532 012737 100000 001124 MOV #ATA, $GDDAT ; WHAT REGISTER SHOULD BE
(2) 027540 013737 001126 001164 MOV $BDDAT, $TMP0 ; MOVE REGISTER CONTENTS TO '$TMP0'
(2) 027546 042737 077777 001164 BIC #1CATA, $TMP0 ; SAVE SPECIFIED BITS
(2) 027554 023737 001124 001164 CMP $GDDAT, $TMP0 ; COMPARE THE BITS
(2) 027562 001414 BEQ 70$ ; BR IF OK
(2) 027564 013737 001126 001174 MOV $BDDAT, $TMP4 ; COPY 'BAD DATA'
(2) 027572 042737 100000 001174 BIC #ATA, $TMP4 ; CLEAR THE MASKED BITS
(2) 027600 053737 001174 001124 BIS $TMP4, $GDDAT ; 'OR' WITH GOOD DATA FOR TYPEOUT
(2) 027606 104016 ERROR 16 ; TYPE MESSAGE 16
(2) 027610 005137 001244 COM 16 ; SET THE REGISTER COMPARE ERROR INDICATOR
(2) 027614 000240 NOP
70$:
(1) ; *****
(2) ; RELEASE THE DRIVE FROM PORT B
(2)
(3) 027616 113760 001226 000010 MOVB PORTB, RPCS2(RO) ; SELECT PORT B
(3) 027624 013737 001226 001234 MOV PORTB, PTNBR ; MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
(3) 027632 012760 000013 000000 MOV #13, RPCS1(RO) ; ISSUE RELEASE THROUGH PORT B
(3) ; VERIFY THAT THE DRIVE IS IN NEUTRAL
(3)
(3) 027640 005037 001250 CLR RELERR ; CLEAR THE 'RELEASE ERROR' INDICATOR
(3) 027644 012737 000012 001122 MOV #RPDS1, $BDAOR ; FORM THE ADDRESS OF RPDS1 FOR TYPEOUT
(3) 027652 060037 001122 ADD RO, $BDAOR ; ADD THE I/O BASE ADDRESS
(3) 027656 012737 011700 001124 MOV #MOL!PGM!DPR!DRY!VV, $GDDAT ; COMPARISON CONSTANT
(3) 027664 113760 001224 000010 MOVB PORTA, RPCS2(RO) ; SELECT PORT A.
(3) 027672 016037 000012 001170 MOV RPDS1(RO), $TMP2 ; GET THE DRIVE STATUS REGISTER FROM PORT A.
(3) 027700 013737 001170 001164 MOV $TMP2, $TMP0 ; COPY IT INTO '$TMP0'
(3) 027706 042737 100100 001164 BIC #ATA!VV, $TMP0 ; CLEAR PORT DEPENDENT BITS FROM THE COPY
(3) 027714 113760 001226 000010 MOVB PORTB, RPCS2(RO) ; SELECT PORT B.
(3) 027722 016037 000012 001172 MOV RPDS1(RO), $TMP3 ; GET THE DRIVE STATUS REGISTER FROM PORT B.
(3) 027730 013737 001172 001166 MOV $TMP3, $TMP1 ; COPY IT INTO '$TMP1'
(3) 027736 042737 100100 001166 BIC #ATA!VV, $TMP1 ; CLEAR PORT DEPENDENT BITS FROM THE COPY
(3) 027744 023737 001164 001166 CMP $TMP0, $TMP1 ; IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
(3) 027752 001006 BNE 72$ ; BR IF NOT
(3) 027754 005737 001164 TST $TMP0 ; REGISTERS ARE THE SAME: ARE THEY ZERO ?
(3) 027760 001045 BNE 74$ ; BR IF NOT
(3) 027762 104046 ERROR 46 ; REPORT DRIVE NOT IN NEUTRAL OR NOT SEIZED
(3) 027764 000137 030150 JMP 76$ ; BYPASS THE REST OF THE CHECKS
(3) 027770 013737 001170 001126 72$: MOV $TMP2, $BDDAT ; SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
(3) 027776 013737 001226 001234 MOV PORTB, PTNBR ; SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
(3) 030004 113760 001226 000010 MOVB PORTB, RPCS2(RO) ; SELECT PORT B.
(3) 030012 005737 001164 TST $TMP0 ; SEE IF STATUS EQ 0 FROM PORT A.
(3) 030016 001414 BEQ 73$ ; BR IF ZERO
(3) 030020 013737 001224 001234 MOV PORTA, PTNBR ; SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
(3) 030026 013737 001172 001126 MOV $TMP3, $BDDAT ; 'BAD DATA' FOR ERROR TYPE OUT
(3) 030034 113760 001224 000010 MOVB PORTA, RPCS2(RO) ; SELECT PORT A.
(3) 030042 005737 001166 TST $TMP1 ; SEE IF STATUS EQ ZERO FROM PORT B.
(3) 030046 001012 BNE 74$ ; BR IF NOT
(3) 030050 012737 177777 001250 73$: MOV #-1, RELERR ; SET 'RELEASE ERROR' INDICATOR
(3) 030056 012760 000011 000000 MOV #11, RPCS1(RO) ; CLEAR THE DRIVE
(3) 030064 012760 000013 000000 MOV #13, RPCS1(RO) ; RELEASE THE DRIVE
```


H08

CZRJECO, DL CTRLR LGC MACY11 30A(1052
CZRJEC.P11 21-DEC-77 14:19

28-DEC-77 10:17 PAGE 67-5
*23 TEST NO 'PORT REQUEST' WHEN READ RPCS1 THROUGH PORT 'A'

SEQ 0098

```

(2) 030626 113760 001226 000010      MOVB  PORTB,RPCS2(RO)  ;SELECT PORT B
(3) 030634 013737 001226 001234      MOV   PORTB,PTNBR    ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
(2) 030642 012760 000013 000000      MOV   #13,RPCS1(RO)  ;ISSUE RELEASE THROUGH PORT B
(3)                                     ;VERIFY THAT THE DRIVE IS IN NEUTRAL
(3) 030650 005037 001250                CLR   RELERR         ;CLEAR THE 'RELEASE ERROR' INDICATOR
(3) 030654 012737 000012 001122      MOV   #RPDS1,$BDDADR ;FORM THE ADDRESS OF RPDS1 FOR TYPEOUT
(3) 030662 060037 001122                ADD   RO,$BDDADR     ;ADD THE I/O BASE ADDRESS
(3) 030666 012737 011700 001124      MOV   #MOL:PGM:DPR:DRY:VV,$GDDAT ;COMPARISON CONSTANT
(3) 030674 113760 001224 000010      MOVB  PORTA,RPCS2(RO) ;SELECT PORT A.
(3) 030702 016037 000012 001170      MOV   RPDS1(RO),STMP2 ;GET THE DRIVE STATUS REGISTER FROM PORT A.
(3) 030710 013737 001170 001164      MOV   STMP2,STMP0    ;COPY IT INTO 'STMP0'
(3) 030716 042737 100100 001164      BIC   #ATA:VV,STMP0  ;CLEAR PORT DEPENDENT BITS FROM THE COPY
(3) 030724 113760 001226 000010      MOVB  PORTB,RPCS2(RO) ;SELECT PORT B.
(3) 030732 016037 000012 001172      MOV   RPDS1(RO),STMP3 ;GET THE DRIVE STATUS REGISTER FROM PORT B.
(3) 030740 013737 001172 001166      MOV   STMP3,STMP1    ;COPY IT INTO 'STMP1'
(3) 030746 042737 100100 001166      BIC   #ATA:VV,STMP1  ;CLEAR PORT DEPENDENT BITS FROM THE COPY
(3) 030754 023737 001164 001166      CMP   STMP0,STMP1    ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
(3) 030762 001006 001164                BNE   68$           ;BR IF NOT
(3) 030764 005737 001164                TST   STMP0         ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
(3) 030770 001045 001164                BNE   70$           ;BR IF NOT
(3) 030772 104046 001164                ERROR 46           ;REPORT DRIVE NOT IN NEUTRAL OR NOT SEIZED
(3) 030774 000137 031160 001126      JMP   72$           ;BYPASS THE REST OF THE CHECKS
(3) 031000 013737 001170 001126 68$:  MOV   STMP2,$BDDAT   ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
(3) 031006 013737 001226 001234      MOV   PORTB,PTNBR    ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
(3) 031014 113760 001226 000010      MOVB  PORTB,RPCS2(RO) ;SELECT PORT B.
(3) 031022 005737 001164                TST   STMP0         ;SEE IF STATUS EQ 0 FROM PORT A.
(3) 031026 001414 001164                BEQ   69$           ;BR IF ZERO
(3) 031030 013737 001224 001234      MOV   PORTA,PTNBR    ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
(3) 031036 013737 001172 001126      MOV   STMP3,$BDDAT   ;'BAD DATA' FOR ERROR TYPE OUT
(3) 031044 113760 001224 000010      MOVB  PORTA,RPCS2(RO) ;SELECT PORT A.
(3) 031052 005737 001166                TST   STMP1         ;SEE IF STATUS EQ ZERO FROM PORT B.
(3) 031056 001012 001166                BNE   70$           ;BR IF NOT
(3) 031060 012737 177777 001250 69$:  MOV   #-1,RELERR     ;SET 'RELEASE ERROR' INDICATOR
(3) 031066 012760 000011 000000      MOV   #11,RPCS1(RO) ;CLEAR THE DRIVE
(3) 031074 012760 000013 000000      MOV   #13,RPCS1(RO) ;RELEASE THE DRIVE
(3) 031102 104026 001170 001126 70$:  ERROR 26           ;TYPE ERROR MESSAGE 26
(3) 031104 013737 001170 001126      MOV   STMP2,$BDDAT   ;LOOK FOR BIT FAILURES WHEN RPDS1 READ
(3) 031112 013737 001224 001234      MOV   PORTA,PTNBR    ;CHANGE PORT NUMBER
(3) 031120 023737 001124 001170      CMP   $GDDAT,STMP2   ;ALL BITS OK ?
(3) 031126 001401 001126                BEQ   71$           ;BR IF OK FROM PORT A.
(3) 031130 104007 001172 001126 71$:  ERROR 7            ;REPORT ERROR
(3) 031132 013737 001172 001126      MOV   STMP3,$BDDAT   ;CHECK RPDS1 FOR BIT FAILURES - FROM PORT B.
(3) 031140 013737 001226 001234      MOV   PORTB,PTNBR    ;CHANGE PORT NUMBER
(3) 031146 023737 001124 001172      CMP   $GDDAT,STMP3   ;SEE IF READ OK FROM PORT B.
(3) 031154 001401 001172                BEQ   72$           ;BR IF OK
(3) 031156 104007 001172                ERROR 7            ;REPORT ERROR
(1) 031160 000240 001172                NOP
(1) 031162 000004 001172                JS    1$           ;LOOP ?

```

BE87
BE88
3
4

```

;*****
;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.

↑ST24:

```

(1) 031164          005737 001274          TST      KYBCTL      ;PERFORMING ONLY SINGLE TESTS
(2) 031164          005737 001274          BEQ      2$          ;BR IF NOT
(3) 031170          001406          BPL      1$          ;BR IF JUST ENTERED TEST
(3) 031172          100002          JMP      EXEC        ;RETURN & GET NEXT TEST NUMBER
(3) 031174          000137 002622          MOV      #-1,KYBCTL  ;SET SINGLE TEST INDICATOR
(3) 031200          012737 177777 001274 1$:      MOV      #24,$STNM   ;TEST NUMBER
(3) 031206          112737 000024 001102 2$:      MOV      #TEST24,$LPADR ;LOAD LOOP ON TEST ADDRESS
(3) 031214          012737 031236 001106      MOV      #TEST24,$LPERR ;LOAD LOOP ON ERROR ADDRESS
(3) 031222          012737 031236 001110      MOV      #4000,$TIMES ;DO 4000. ITERATIONS
(1) 031230          012737 007640 001176      MOV      #STACK,$SP  ;LOAD THE STACK POINTER
(3) 031236          012706 001100          TEST24: MOV      #STACK,$SP  ;LOAD THE STACK POINTER

(2)          ;CLEAR ATTENTION BITS FOR BOTH PORTS
(2) 031242          113760 001224 000010      MOV      PORTA,RPCS2(RO) ;SELECT PORT #A
(2) 031250          005060 000012          CLR      RPDS1(RO)      ;SEIZE THE DRIVE
(2) 031254          012760 000011 000000      MOV      #11,RPCS1(RO) ;ISSUE DRIVE CLEAR
(2) 031262          012760 000013 000000      MOV      #13,RPCS1(RO) ;RELEASE THE DRIVE
(2) 031270          113760 001226 000010      MOV      PORTB,RPCS2(RO) ;SELECT PORT #B
(2) 031276          005060 000012          CLR      RPDS1(RO)      ;SEIZE THE DRIVE THROUGH PORT 'B'
(2) 031302          012760 000011 000000      MOV      #11,RPCS1(RO) ;ISSUE DRIVE CLEAR
(2) 031310          012760 000013 000000      MOV      #13,RPCS1(RO) ;RELEASE THE DRIVE

(1)          ;*****
(2)          ;SEIZE THE DRIVE THROUGH PORT A
(2) 031316          113760 001224 000010      MOV      PORTA,RPCS2(RO) ;SELECT PORT A
(2) 031324          013737 001224 001236      MOV      PORTA,SEIZPT ;STORE SEIZING PORT'S ADDRESS
(2) 031332          005760 000000          TST      RPCS1(RO)      ;READ RPCS1
(2) 031336          113760 001226 000010      MOV      PORTB,RPCS2(RO) ;SELECT PORT B
(2) 031344          013737 001226 001234      MOV      PORTB,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
(2) 031352          013737 001226 001240      MOV      PORTB,OPPRT ;'OPPOSITE' PORT ADDRESS
(2) 031360          016037 000012 001126      MOV      RPDS1(RO),$BDDAT ;SEE IF DRIVE SEIZED BY PORT A
(2) 031366          010037 001122          MOV      RO,$BDADR      ;R#11 BASE ADDRESS
(2) 031372          062737 000012 001122      ADD      #RPDS1,$BDADR  ;GENERATE BAD REGISTER ADDRESS
(2) 031400          005037 001124          CLR      $GDDAT        ;REGISTER SHOULD BE ZERO
(2) 031404          023737 001124 001126      CMP      $GDDAT,$BDDAT ;IS THE REGISTER ZERO
(2) 031412          001403          BEQ      64$          ;BR IF IT IS
(2) 031414          104004          ERROR    4            ;REPORT THE ERROR
(2) 031416          000137 032172          JMP      1$            ;BYPASS REST OF THE SUBTEST
(2) 031422          001403          BEQ      64$          ;BR IF IT IS
(3) 031422          113760 001224 000010 64$:      MOV      PORTA,RPCS2(RO) ;SELECT PORT A
(3) 031430          013737 001224 001236      MOV      PORTA,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT

```

J08

```
(R) 031436 016037 000012 001126 MOV RPDS1(RO) $BDDAT ;SEE IF SEIZING PORT SEES CORRECT STATUS
(R) 031444 012737 011700 001124 MOV #MOL!PGM!DPR!DRY!VV,$GDDAT ;EXPECTED STATUS
(R) 031452 013737 001124 001166 MOV $GDDAT,$TMP1 ;USE GOOD DATA AS A MASK
(R) 031460 005137 001166 COM $TMP1 ;COMPLEMENT THE EXPECTED STATUS
(R) 031464 013737 001126 001164 MOV $BDDAT,$TMP0 ;SAVE THE ACTUAL STATUS
(R) 031472 043737 001166 001164 BIC $TMP1,$TMP0 ;CLEAR UNWANTED BITS
(R) 031500 023737 001124 001164 CMP $GDDAT,$TMP0 ;ARE THE EXPECTED STATUS BITS SET?
(R) 031506 001401 BEQ 65$ ;BR IF THEY ARE
(R) 031510 104005 ERROR 5 ;REPORT THE ERROR
(R) 031512 000240 65$: NOP
(R) 031514 113760 001226 000010 MOVB PORTB,RPCS2(RO) ;SELECT PORT B
(R) 031522 013737 001226 001234 MOV PORTB,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
```

 ;READ RPCS1 THROUGH PORT B - TRY TO SET PORT REQUEST

```
(R) 031530 005037 001244 CLR CKERR ;CLEAR THE 'CHECK ERROR' INDICATOR
(R) 031534 016037 000000 001126 MOV RPCS1(RO) $BDDAT ;GET CONTENTS OF RPCS1
(R) 031542 012737 000000 001122 MOV #RPCS1,$BDAOR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
(R) 031550 060037 001122 ADD RO,$BDAOR ;ADD RHI1 BASE ADDRESS
(R) 031554 005037 001124 CLR $GDDAT ;WHAT REGISTER SHOULD BE
(R) 031560 013737 001126 001164 MOV $BDDAT,$TMP0 ;MOVE REGISTER CONTENTS TO '$TMP0'
(R) 031566 042737 173700 001164 BIC #14077,$TMP0 ;SAVE SPECIFIED BITS
(R) 031574 023737 001124 001164 CMP $GDDAT,$TMP0 ;COMPARE THE BITS
(R) 031602 001414 BEQ 66$ ;BR IF OK
(R) 031604 013737 001126 001174 MOV $BDDAT,$TMP4 ;COPY 'BAD DATA'
(R) 031612 042737 004077 001174 BIC #4077,$TMP4 ;CLEAR THE MASKED BITS
(R) 031620 053737 001174 001124 BIS $TMP4,$GDDAT ;'OR' WITH GOOD DATA FOR TYPEOUT
(R) 031626 104010 ERROR 10 ;REPORT THE ERROR
(R) 031630 005137 001244 COM CKERR ;SET THE REGISTER COMPARE ERROR INDICATOR
(R) 031634 000240 66$: NOP
```

 ;DRIVE SHOULD RETURN TO NEUTRAL

;RELEASE THE DRIVE FROM PORT A

```
(R) 031636 113760 001224 000010 MOVB PORTA,RPCS2(RO) ;SELECT PORT A
(R) 031644 013737 001224 001234 MOV PORTA,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
(R) 031652 012760 000013 000000 MOV #13,RPCS1(RO) ;ISSUE RELEASE THROUGH PORT A
```

;VERIFY THAT THE DRIVE IS IN NEUTRAL

```
(R) 031660 005037 001250 CLR RELERR ;CLEAR THE 'RELEASE ERROR' INDICATOR
(R) 031664 012737 000012 001122 MOV #RPDS1,$BDAOR ;FORM THE ADDRESS OF RPDS1 FOR TYPEOUT
(R) 031672 060037 001122 ADD RO,$BDAOR ;ADD THE I/O BASE ADDRESS
(R) 031676 012737 011700 001124 MOV #MOL!PGM!DPR!DRY!VV,$GDDAT ;COMPARISON CONSTANT
(R) 031704 113760 001224 000010 MOVB PORTA,RPCS2(RO) ;SELECT PORT A.
(R) 031712 016037 000012 001170 MOV RPLS1(RO) $TMP2 ;GET THE DRIVE STATUS REGISTER FROM PORT A.
(R) 031720 013737 001170 001164 MOV $TMP2,$TMP0 ;COPY IT INTO '$TMP0'
(R) 031726 042737 100100 001164 BIC #ATA!VV,$TMP0 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
(R) 031734 113760 001226 000010 MOVB PORTB,RPCS2(RO) ;SELECT PORT B.
(R) 031742 016037 000012 001172 MOV RPDS1(RO) $TMP3 ;GET THE DRIVE STATUS REGISTER FROM PORT B.
(R) 031750 013737 001172 001166 MOV $TMP3,$TMP1 ;COPY IT INTO '$TMP1'
(R) 031756 042737 100100 001166 BIC #ATA!VV,$TMP1 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
```



```

(3) 032202 100002          BPL      1$          ;BR IF JUST ENTERED TEST
(3) 032204 000137 002622    JMP      EXEC        ;RETURN & GET NEXT TEST NUMBER
(3) 032210 012737 177777 001274 1$: MOV     #1,KYBCTL    ;SET SINGLE TEST INDICATOR
(3) 032216 112737 000025 001102 2$: MOV     #25,$TSTNM ;TEST NUMBER
(3) 032224 012737 032246 001106    MOV     #TEST25,$LPADR ;LOAD LOOP ON TEST ADDRESS
(3) 032232 012737 032246 001110    MOV     #TEST25,$LPERR ;LOAD LOOP ON ERROR ADDRESS
(1) 032240 012737 007640 001176    MOV     #4000,$TIMES   ;DO 4000. ITERATIONS
8712 032246 012706 001100    TEST25: MOV    #STACK,$P ;LOAD THE STACK POINTER
8738
(2)                               ;CLEAR ATTENTION BITS FOR BOTH PORTS
(2) 032252 113760 001224 000010    MOV     PORTA,RPCS2(RO) ;SELECT PORT #A
(2) 032260 005060 000012          CLR     RPDS1(RO)      ;SEIZE THE DRIVE
(2) 032264 012760 000011 000000    MOV     #11,RPCS1(RO)  ;ISSUE DRIVE CLEAR
(2) 032272 012760 000013 000000    MOV     #13,RPCS1(RO)  ;RELEASE THE DRIVE
(2) 032300 113760 001226 000010    MOV     PORTB,RPCS2(RO) ;SELECT PORT #B
(2) 032306 005060 000012          CLR     RPDS1(RO)      ;SEIZE THE DRIVE THROUGH PORT 'B'
(2) 032312 012760 000011 000000    MOV     #11,RPCS1(RO)  ;ISSUE DRIVE CLEAR
(2) 032320 012760 000013 000000    MOV     #13,RPCS1(RO)  ;RELEASE THE DRIVE
(1)
(2)                               ;*****
(2)                               ;SEIZE THE DRIVE THROUGH PORT B
(2) 032326 113760 001226 000010    MOV     PORTB,RPCS2(RO) ;SELECT PORT B
(2) 032334 013737 001226 001236    MOV     PORTB,SEIZPT   ;STORE SEIZING PORT'S ADDRESS
(2) 032342 005060 000012          CLR     RPDS1(RO)      ;WRITE RPDS1
(3) 032346 113760 001224 000010    MOV     PORTA,RPCS2(RO) ;SELECT PORT A
(3) 032354 013737 001224 001234    MOV     PORTA,PTNBR    ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
(2) 032362 013737 001224 001240    MOV     PORTA,OPPR    ;'OPPOSITE' PORT ADDRESS
(2) 032370 016037 000012 001126    MOV     RPDS1(RO),$BDDAT ;SEE IF DRIVE SEIZED BY PORT B
(2) 032376 010037 001122          MOV     RO,$BDAADR    ;R#11 BASE ADDRESS
(2) 032402 062737 000012 001122    ADD     #RPDS1,$BDAADR ;GENERATE BAD REGISTER ADDRESS
(2) 032410 005037 001124          CLR     $GDDAT        ;REGISTER SHOULD BE ZERO
(2) 032414 023737 001124 001126    CMP     $GDDAT,$BDDAT ;IS THE REGISTER ZERO
(2) 032422 001403          BEQ     64$          ;BR IF IT IS
(2) 032424 104004          ERROR  4            ;REPORT THE ERROR
(2) 032426 000137 033400          JMP     1$          ;BYPASS REST OF THE SUBTEST
(2) 032432
(3) 032432 113760 001226 000010 64$: MOV     PORTB,RPCS2(RO) ;SELECT PORT B
(3) 032440 013737 001226 001234    MOV     PORTB,PTNBR    ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
(2) 032446 016037 000012 001126    MOV     RPDS1(RO),$BDDAT ;SEE IF SEIZING PORT SEES CORRECT STATUS
(2) 032454 012737 011700 001124    MOV     #MOL:PGM:OPR:DRY:VV,$GDDAT ;EXPECTED STATUS
(2) 032462 013737 001124 001166    MOV     $GDDAT,$TMP1  ;USE GOOD DATA AS A MASK
(2) 032470 005137 001166          COM     $TMP1         ;COMPLEMENT THE EXPECTED STATUS
(2) 032474 013737 001126 001164    MOV     $BDDAT,$TMP0  ;SAVE THE ACTUAL STATUS
(2) 032502 043737 001166 001164    BIC     $TMP1,$TMP0   ;CLEAR UNWANTED BITS
(2) 032510 023737 001124 001164    CMP     $GDDAT,$TMP0  ;ARE THE EXPECTED STATUS BITS SET ?
(2) 032516 001401          BEQ     65$          ;BR IF THEY ARE
(2) 032520 104005          ERROR  5            ;REPORT THE ERROR
(2) 032522 000240 65$: NOP
(1)
(2)                               ;*****
(1)                               ;TRY TO EXECUTE A RELEASE COMMAND THROUGH PORT A
(1)
(2) 032524 113760 001224 000010    MOV     PORTA,RPCS2(RO) ;SELECT PORT A

```

```

(2) 032532 013737 001224 001234      MOV      PORTA,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
(1) 032540 012760 000013 000000      MOV      #13,RPCS1(RO) ;ISSUE A RELEASE COMMAND THROUGH PORT A
(1)
(2) ;*****
(1) ;VERIFY THAT THE DRIVE IS STILL SEIZED BY PORT B
(1)
(2) 032546 005037 001244          CLR      CKERR ;CLEAR THE 'CHECK ERROR' INDICATOR
(2) 032552 016037 000012 001126      MOV      RPS1(RO), $BDDAT ;GET CONTENTS OF RPS1
(2) 032560 012737 000012 001122      MOV      #RPS1,$BDAOR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
(2) 032566 060037 001122          ADD      RO,$BDAOR ;ADD RHI1 BASE ADDRESS
(2) 032572 005037 001124          CLR      $GDDAT ;WHAT REGISTER SHOULD BE
(2) 032576 023737 001124 001126      CMP      $GDDAT,$BDDAT ;IS THE REGISTER OK?
(2) 032604 001403          BEQ      66$ ;BR IF OK
(2) 032606 104010          ERROR    10 ;REPORT THE ERROR
(2) 032610 005137 001244          COM      CKERR ;SET THE REGISTER COMPARE ERROR INDICATOR
(2) 032614 016037 000000 001126 66$: MOV      RPCS1(RO), $BDDAT ;GET THE CONTENTS OF RPS1
(2) 032622 012737 000000 001122      MOV      #RPCS1,$BDAOR ;FORM ADDRESS OF REGISTER
(2) 032630 060037 001122          ADD      RO,$BDAOR ;ADDRESS BASE
(2) 032634 032737 020000 001126      BIT      #MCPE,$BDDAT ;IS 'MCPE' SET?
(2) 032642 001404          BEQ      67$ ;BR IF NOT
(2) 032644 104011          ERROR    11 ;REPORT THE ERROR
(2) 032646 012760 040000 000000      MOV      #TRE,RPCS1(RO) ;CLEAR 'MCPE'
(2) 032654 000240          NOP
(1) 032656 005737 001244          TST      CKERR ;WAS RPS1 NON ZERO?
(1) 032662 001402          BEQ      +6 ;CONTENTS OF RPS1 SEEN BY PORT A
(1) 022664 000137 033400          JMP      1$ ;DRIVE IN NEUTRAL, BYPASS REST OF TEST
(1)
(2) ;*****
(2) ;RELEASE THE DRIVE FROM PORT B
(2)
(3) 032670 113760 001226 000010      MOV      PORTB,RPCS2(RO) ;SELECT PORT B
(3) 032676 013737 001226 001234      MOV      PORTB,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
(2) 032704 012760 000013 000000      MOV      #13,RPCS1(RO) ;ISSUE RELEASE THROUGH PORT B
(3)
(3) ;VERIFY THAT DRIVE IS SEIZED BY PORT A WHEN RELEASED BY PORT B
(3)
(3) 032712 005037 001250          CLR      RELERR ;CLEAR 'RELEASE ERROR' INDICATOR
(3) 032716 012737 111700 001124      MOV      #ATA!MOL!PGM!DPR!DRY!VV,$GDDAT ;COMPARISON CONSTANT
(3) 032724 012737 000012 001122      MOV      #RPS1,$BDAOR ;REGISTER ADDRESS INCREMENT
(3) 032732 060037 001122          ADD      RO,$BDAOR ;REGISTER BASE ADDRESS FOR TYPEOUT
(4) 032736 113760 001224 000010      MOV      PORTA,RPCS2(RO) ;SELECT PORT A
(4) 032744 013737 001224 001234      MOV      PORTA,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
(2) 032752 016037 000012 001164      MOV      RPS1(RO), $TMPD ;READ STATUS REGISTER FROM PORT A
(4) 032760 113760 001226 000010      MOV      PORTB,RPCS2(RO) ;SELECT PORT B
(4) 032766 013737 001226 001234      MOV      PORTB,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
(3) 032774 016037 000012 001126      MOV      RPS1(RO), $BDDAT ;DRIVE STATUS FROM PORT B
(3) 033002 001404          BEQ      68$ ;BR IF STATUS FROM PORT B ZERO
(3) 033004 005737 001164          TST      $TMPD ;IS STATUS FROM PORT A ZERO?
(3) 033010 001401          BEQ      68$ ;BR IF ZERO
(3) 033012 104031          ERROR    31 ;REPORT DRIVE IN NEUTRAL
(3) 033014 013737 001164 001126 68$: MOV      $TMPD,$BDDAT ;CHECK STATUS FROM PORT A
(3) 033022 013737 001224 001234      MOV      PORTA,PTNBR ;CHANGE PORT ADDRESS FOR TYPEOUT
(3) 033030 023737 001124 001126      CMP      $GDDAT,$BDDAT ;COMPARE WITH CONSTANT
(3) 033036 001401          BEQ      69$ ;BR IF OK
(3) 033040 104027          ERROR    27 ;REPORT REGISTER ERROR

```



```

(3) 033042 000240 69$: NOP
(2) ;RELEASE THE DRIVE FROM PORT A
(2)
(3) 033044 113760 001224 000010 MOVB PORTA,RPCS2(RO) ;SELECT PORT A
(3) 033052 013737 001224 001234 MOV PORTA,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
(2) 033060 012760 000013 000000 MOV #13,RPCS1(RO) ;ISSUE RELEASE THROUGH PORT A
(3)
(3) ;VERIFY THAT THE DRIVE IS IN NEUTRAL
(3)
(3) 033066 005037 001250 CLR RELERR ;CLEAR THE 'RELEASE ERROR' INDICATOR
(3) 033072 012737 000012 001122 MOV #RPDS1,$BDDADR ;FORM THE ADDRESS OF RPDS1 FOR TYPEOUT
(3) 033100 060037 001122 ADD RO,$BDDADR ;ADD THE I/O BASE ADDRESS
(3) 033104 012737 011700 001124 MOV #MOL!PGM!DPR!DRY!VV,$GDDAT ;COMPARISON CONSTANT
(3) 033112 113760 001224 000010 MOVB PORTA,RPCS2(RO) ;SELECT PORT A.
(3) 033120 016037 000012 001170 MOV RPDS1(RO),$TMP2 ;GET THE DRIVE STATUS REGISTER FROM PORT A.
(3) 033126 013737 001170 001164 MOV $TMP2,$TMP0 ;COPY IT INTO 'TMP0'
(3) 033134 042737 100100 001164 BIC #ATA!VV,$TMP0 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
(3) 033142 113760 001226 000010 MOVB PORTB,RPCS2(RO) ;SELECT PORT B.
(3) 033150 016037 000012 001172 MOV RPDS1(RO),$TMP3 ;GET THE DRIVE STATUS REGISTER FROM PORT B.
(3) 033156 013737 001172 001166 MOV $TMP3,$TMP1 ;COPY IT INTO 'TMP1'
(3) 033164 042737 100100 001166 BIC #ATA!VV,$TMP1 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
(3) 033172 023737 001164 001166 CMP $TMP0,$TMP1 ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
(3) 033200 001006 BNE 70$ ;BR IF NOT
(3) 033202 005737 001164 TST $TMP0 ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
(3) 033206 001045 BNE 72$ ;BR IF NOT
(3) 033210 104046 ERROR 46 ;REPORT DRIVE NOT IN NEUTRAL OR NOT SEIZED
(3) 033212 000137 033376 JMP 74$ ;BYPASS THE REST OF THE CHECKS
(3) 033216 013737 001170 001126 70$: MOV $TMP2,$BDDAT ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
(3) 033224 013737 001226 001234 MOV PORTB,PTNBR ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
(3) 033232 113760 001226 000010 MOVB PORTB,RPCS2(RO) ;SELECT PORT B.
(3) 033240 005737 001164 TST $TMP0 ;SEE IF STATUS EQ 0 FROM PORT A.
(3) 033244 001414 BEQ 71$ ;BR IF ZERO
(3) 033246 013737 001224 001234 MOV PORTA,PTNBR ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
(3) 033254 013737 001172 001126 MOV $TMP3,$BDDAT ;'BAD DATA' FOR ERROR TYPE OUT
(3) 033262 113760 001224 000010 MOVB PORTA,RPCS2(RO) ;SELECT PORT A.
(3) 033270 005737 001166 TST $TMP1 ;SEE IF STATUS EQ ZERO FROM PORT B.
(3) 033274 001012 BNE 72$ ;BR IF NOT
(3) 033276 012737 177777 001250 71$: MOV #-1,RELERR ;SET 'RELEASE ERROR' INDICATOR
(3) 033304 012760 000011 000000 MOV #11,RPCS1(RO) ;CLEAR THE DRIVE
(3) 033312 012760 000013 000000 MOV #13,RPCS1(RO) ;RELEASE THE DRIVE
(3) 033320 104026 ERROR 26 ;TYPE ERROR MESSAGE 26
(3) 033322 013737 001170 001126 72$: MOV $TMP2,$BDDAT ;LOOK FOR BIT FAILURES WHEN RPDS1 READ
(3) 033330 013737 001224 001234 MOV PORTA,PTNBR ;CHANGE PORT NUMBER
(3) 033336 023737 001124 001170 CMP $GDDAT,$TMP2 ;ALL BITS OK ?
(3) 033344 001401 BEQ 73$ ;BR IF OK FROM PORT A.
(3) 033346 104007 ERROR 7 ;REPORT ERROR
(3) 033350 013737 001172 001126 73$: MOV $TMP3,$BDDAT ;CHECK RPDS1 FOR BIT FAILURES - FROM PORT B.
(3) 033356 013737 001226 001234 MOV PORTB,PTNBR ;CHANGE PORT NUMBER
(3) 033364 023737 001124 001172 CMP $GDDAT,$TMP3 ;SEE IF READ OK FROM PORT B.
(3) 033372 001401 BEQ 74$ ;BR IF OK
(3) 033374 104007 ERROR 7 ;REPORT ERROR
(3) 033376 000240 74$: NOP
(1) 033400 000004 1$: SCOPE ;LOOP ?

```

8757
8758

::*****


```

(2) 033630 001403          BEQ      64$          ;BR IF IT IS
(2) 033632 104004          ERROR    4           ;REPORT THE ERROR
(2) 033634 000137 034606   JMP      1$          ;BYPASS REST OF THE SUBTEST
(2) 033640
(3) 033640 113760 001224 000010 64$:  MOVB    PORTA,RPCS2(RO) ;SELECT PORT A
(2) 033646 013737 001224 001234   MOV     PORTA,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
(2) 033654 016037 000012 001126   MOV     RPDS1(RO), $BDDAT ;SEE IF SEIZING PORT SEES CORRECT STATUS
(2) 033662 012737 011700 001124   MOV     #MOL!PGM!DPR!DRY!VV $GDDAT ;EXPECTED STATUS
(2) 033670 013737 001124 001166   MOV     $GDDAT, $TMP1 ;USE GOOD DATA AS A MASK
(2) 033676 005137 001166          COM     $TMP1 ;COMPLEMENT THE EXPECTED STATUS
(2) 033702 013737 001126 001164   MOV     $BDDAT, $TMP0 ;SAVE THE ACTUAL STATUS
(2) 033710 043737 001166 001164   BIC     $TMP1, $TMP0 ;CLEAR UNWANTED BITS
(2) 033716 023737 001124 001164   CMP     $GDDAT, $TMP0 ;ARE THE EXPECTED STATUS BITS SET ?
(2) 033724 001401          BEQ     65$          ;BR IF THEY ARE
(2) 033726 104005          ERROR   5           ;REPORT THE ERROR
(2) 033730 000240          65$:  NOP

;*****
;TRY TO EXECUTE A RELEASE COMMAND THROUGH PORT B
(2) 033732 113760 001226 000010   MOVB    PORTB,RPCS2(RO) ;SELECT PORT B
(2) 033740 013737 001226 001234   MOV     PORTB,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
(1) 033746 012760 000013 000000   MOV     #13,RPDS1(RO) ;ISSUE A RELEASE COMMAND THROUGH PORT B
(1)
(2)
(1)
;*****
;VERIFY THAT THE DRIVE IS STILL SEIZED BY PORT A
(2) 033754 005037 001244          CLR     CKERR ;CLEAR THE 'CHECK ERROR' INDICATOR
(2) 033760 016037 000012 001126   MOV     RPDS1(RO), $BDDAT ;GET CONTENTS OF RPDS1
(2) 033766 012737 000012 001122   MOV     #RPDS1, $B0ADR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
(2) 033774 060037 001122          ADD     RO, $B0ADR ;ADD RH11 BASE ADDRESS
(2) 034000 005037 001124          CLR     $GDDAT ;WHAT REGISTER SHOULD BE
(2) 034004 023737 001124 001126   CMP     $GDDAT, $BDDAT ;IS THE REGISTER OK ?
(2) 034012 001403          BEQ     66$          ;BR IF OK
(2) 034014 104010          ERROR   10         ;REPORT THE ERROR
(2) 034016 005137 001244          COM     CKERR ;SET THE REGISTER COMPARE ERROR INDICATOR
(2) 034022 016037 000000 001126 66$:  MOV     RPDS1(RO), $BDDAT ;GET THE CONTENTS OF RHCS1
(2) 034030 012737 000000 001122   MOV     #RPDS1, $B0ADR ;FORM ADDRESS OF REGISTER
(2) 034036 060037 001122          ADD     RO, $B0ADR ;ADDRESS BASE
(2) 034042 032737 020000 001126   BIT     #MCPE, $BDDAT ;IS 'MCPE' SET ?
(2) 034050 001404          BEQ     67$          ;BR IF NOT
(2) 034052 104011          ERROR   11         ;REPORT THE ERROR
(2) 034054 012760 040000 000000   MOV     #TRE, RPDS1(RO) ;CLEAR 'MCPE'
(2) 034062 000240          67$:  NOP
(1) 034064 005737 001244          TST     CKERR ;WAS RPDS1 NON ZERO ?
(1) 034070 001402          BEQ     +6 ;CONTENTS OF RPDS1 SEEN BY PORT B
(1) 034072 000137 034606          JMP     1$ ;DRIVE IN NEUTRAL, BYPASS REST OF TEST
(1)
(2)
;*****
;RELEASE THE DRIVE FROM PORT A
(3) 034076 113760 001224 000010   MOVB    PORTA,RPCS2(RO) ;SELECT PORT A
(3) 034104 013737 001224 001234   MOV     PORTA,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
(3) 034112 012760 000013 000000   MOV     #13,RPDS1(RO) ;ISSUE RELEASE THROUGH PORT A
(3)

```

```

(3) ;VERIFY THAT DRIVE IS SEIZED BY PORT B WHEN RELEASED BY PORT A
(3)
(3) 034120 005037 001250 CLR RELERR ;CLEAR 'RELEASE ERROR' INDICATOR
(3) 034124 012737 111700 001124 MOV #ATA!MOL!PGM!DPR!DRY!VV,$GDDAT ;COMPARISON CONSTANT
(3) 034132 012737 000012 001122 MOV #RPS1,$BDAOR ;REGISTER ADDRESS INCREMENT
(3) 034140 060037 001122 ADD RO,$BDAOR ;REGISTER BASE ADDRESS FOR TYPEOUT
(4) 034144 113760 001226 000010 MOV#B PORTB,RPCS2(RO) ;SELECT PORT B
(4) 034152 013737 001226 001234 MOV PORTB,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
(3) 034160 016037 000012 001164 MOV RPS1(RO),$TMP0 ;READ STATUS REGISTER FROM PORT B
(4) 034166 113760 001224 000010 MOV#B PORTA,RPCS2(RO) ;SELECT PORT A
(4) 034174 013737 001224 001234 MOV PORTA,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
(3) 034202 016037 000012 001126 MOV RPS1(RO),$BDDAT ;DRIVE STATUS FROM PORT A
(3) 034210 001404 BEQ 68$ ;BR IF STATUS FROM PORT A ZERO
(3) 034212 005737 001164 TST $TMP0 ;IS STATUS FROM PORT B ZERO ?
(3) 034216 001401 BEQ 68$ ;BR IF ZERO
(3) 034220 104031 ERROR 31 ;REPORT DRIVE IN NEUTRAL
(3) 034222 013737 001164 001126 68$: MOV $TMP0,$BDDAT ;CHECK STATUS FROM PORT B
(3) 034230 013737 001226 001234 MOV PORTB,PTNBR ;CHANGE PORT ADDRESS FOR TYPEOUT
(3) 034236 023737 001124 001126 CMP $GDDAT,$BDDAT ;COMPARE WITH CONSTANT
(3) 034244 001401 BEQ 69$ ;BR IF OK
(3) 034246 104027 ERROR 27 ;REPORT REGISTER ERROR
(3) 034250 000240 69$: NOP

;RELEASE THE DRIVE FROM PORT B
(3)
(3) 034252 113760 001226 000010 MOV#B PORTB,RPCS2(RO) ;SELECT PORT B
(3) 034260 013737 001226 001234 MOV PORTB,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
(3) 034266 012760 000013 000000 MOV #13,RPCS1(RO) ;ISSUE RELEASE THROUGH PORT B

;VERIFY THAT THE DRIVE IS IN NEUTRAL
(3)
(3) 034274 005037 001250 CLR RELERR ;CLEAR THE 'RELEASE ERROR' INDICATOR
(3) 034300 012737 000012 001122 MOV #RPS1,$BDAOR ;FORM THE ADDRESS OF RPS1 FOR TYPEOUT
(3) 034306 060037 001122 ADD RO,$BDAOR ;ADD THE I/O BASE ADDRESS
(3) 034312 012737 011700 001124 MOV #MOL!PGM!DPR!DRY!VV,$GDDAT ;COMPARISON CONSTANT
(3) 034320 113760 001224 000010 MOV#B PORTA,RPCS2(RO) ;SELECT PORT A.
(3) 034326 016037 000012 001170 MOV RPS1(RO),$TMP2 ;GET THE DRIVE STATUS REGISTER FROM PORT A.
(3) 034334 013737 001170 001164 MOV $TMP2,$TMP0 ;COPY IT INTO '$TMP0'
(3) 034342 042737 100100 001164 BIC #ATA!VV,$TMP0 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
(3) 034350 113760 001226 000010 MOV#B PORTB,RPCS2(RO) ;SELECT PORT B.
(3) 034356 016037 000012 001172 MOV RPS1(RO),$TMP3 ;GET THE DRIVE STATUS REGISTER FROM PORT B.
(3) 034364 013737 001172 001166 MOV $TMP3,$TMP1 ;COPY IT INTO '$TMP1'
(3) 034372 042737 100100 001166 BIC #ATA!VV,$TMP1 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
(3) 034400 023737 001164 001166 CMP $TMP0,$TMP1 ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
(3) 034406 001006 BNE 70$ ;BR IF NOT
(3) 034410 005737 001164 TST $TMP0 ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
(3) 034414 001045 BNE 72$ ;BR IF NOT
(3) 034416 104046 ERROR 46 ;REPORT DRIVE NOT IN NEUTRAL OR NOT SEIZED
(3) 034420 000137 034604 JMP 74$ ;BYPASS THE REST OF THE CHECKS
(3) 034424 013737 001170 001126 70$: MOV $TMP2,$BDDAT ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
(3) 034432 013737 001226 001234 MOV PORTB,PTNBR ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
(3) 034440 113760 001226 000010 MOV#B PORTB,RPCS2(RO) ;SELECT PORT B.
(3) 034446 005737 001164 TST $TMP0 ;SEE IF STATUS EQ 0 FROM PORT A.
(3) 034452 001414 BEQ 71$ ;BR IF ZERO
(3) 034454 013737 001224 001234 MOV PORTA,PTNBR ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
(3) 034462 013737 001172 001126 MOV $TMP3,$BDDAT ;'BAD DATA' FOR ERROR TYPE OUT

```

```

(3) 034470 113760 001224 000010      MOVB  PORTA,RPCS2(RO) ;SELECT PORT A.
(3) 034476 005737 001166              TST   $TMP1           ;SEE IF STATUS EQ ZERO FROM PORT B.
(3) 034502 001012                      BNE   72$            ;BR IF NOT
(3) 034504 012737 177777 001250 71$:  MOV   #-1,RELEARR    ;SET 'RELEASE ERROR' INDICATOR
(3) 034512 012760 000011 000000      MOV   #11,RPCS1(RO) ;CLEAR THE DRIVE
(3) 034520 012760 000013 000000      MOV   #13,RPCS1(RO) ;RELEASE THE DRIVE
(3) 034526 104026                      ERROR 26             ;TYPE ERROR MESSAGE 26
(3) 034530 013737 001170 001126 72$:  MOV   $TMP2,$BDDAT   ;LOOK FOR BIT FAILURES WHEN RPOS1 READ
(3) 034536 013737 001224 001234      MOV   PORTA,PTNBR   ;CHANGE PORT NUMBER
(3) 034544 023737 001124 001170      CMP   $GDDAT,$TMP2 ;ALL BITS OK ?
(3) 034552 001401                      BEQ   73$            ;BR IF OK FROM PORT A.
(3) 034554 104007                      ERROR 7              ;REPORT ERROR
(3) 034556 013737 001172 001126 73$:  MOV   $TMP3,$BDDAT ;CHECK RPOS1 FOR BIT FAILURES - FROM PORT B.
(3) 034564 013737 001226 001234      MOV   PORTB,PTNBR  ;CHANGE PORT NUMBER
(3) 034572 023737 001124 001172      CMP   $GDDAT,$TMP3 ;SEE IF READ OK FROM PORT B.
(3) 034600 001401                      BEQ   74$            ;BR IF OK
(3) 034602 104007                      ERROR 7              ;REPORT ERROR
(3) 034604 000240                      NOP
(1) 034606 000004                      1$:  SCOPE              ;LOOP ?

```

```

*****
*TEST 27      TEST SEIZE BY WRITING ATTENTION BIT
*
*TEST THAT WRITING THE APPROPRIATE 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.
*****

```

```

(3) 034610 005737 001274              TST   KYBCTL         ;PERFORMING ONLY SINGLE TESTS ?
(3) 034614 001406                      BEQ   2$             ;BR IF NOT
(3) 034616 100002                      BPL   1$             ;BR IF JUST ENTERED TEST
(3) 034620 000137 002622              JMP   EXEC           ;RETURN & GET NEXT TEST NUMBER
(3) 034624 012737 177777 001274 1$:  MOV   #-1,KYBCTL    ;SET SINGLE TEST INDICATOR
(3) 034632 112737 000027 001102 2$:  MOVB  #27,$STNM     ;TEST NUMBER
(3) 034640 012737 034662 001106      MOV   #TEST27,$LPADR ;LOAD LOOP ON TEST ADDRESS
(3) 034646 012737 034662 001110      MOV   #TEST27,$LPERR ;LOAD LOOP ON ERROR ADDRESS
(1) 034654 012737 007640 001176      MOV   #4000.,$TIMES ;DO 4000. ITERATIONS
8778 034662 012706 001100 TEST27: MOV   #STACK,SP ;LOAD THE STACK POINTER
8832

```

;CLEAR ATTENTION BITS FOR BOTH PORTS

```

(2) 034666 113760 001224 000010      MOVB  PORTA,RPCS2(RO) ;SELECT PORT #A
(2) 034674 005060 000012              CLR   RPOS1(RO)     ;SEIZE THE DRIVE
(2) 034700 012760 000011 000000      MOV   #11,RPCS1(RO) ;ISSUE DRIVE CLEAR
(2) 034706 012760 000013 000000      MOV   #13,RPCS1(RO) ;RELEASE THE DRIVE
(2) 034714 113760 001226 000010      MOVB  PORTB,RPCS2(RO) ;SELECT PORT #B
(2) 034722 005060 000012              CLR   RPOS1(RO)     ;SEIZE THE DRIVE THROUGH PORT 'B'

```

```

(2) 034726 012760 000011 000000      MOV  #11,RPCS1(RO)  ;ISSUE DRIVE CLEAR
(2) 034734 012760 000013 000000      MOV  #13,RPCS1(RO)  ;RELEASE THE DRIVE
(1)                                     ;*****
(1)                                     ;SELECT DRIVE OTHER THAN THAT BEING TESTED
(1) 034742 113760 001230 000010      MOVB PORTC,RPCS2(RO) ;SELECT DRIVE NOT BEING TESTED
(1) 034750 013737 001224 001236      MOV  PORTA,SEIZPT  ;'SEIZED' PORT ADDRESS
(1)                                     ;*****
(1)                                     ;WRITE THE DRIVE'S ATTENTION BIT
(1) 034756 013760 001232 000016      MOV  ASR1,RPAS(RO) ;WRITE THE ATTENTION BIT OF THE DRIVE BEING TESTED
(2) 034764 113760 001224 000010      MOVB PORTA,RPCS2(RO) ;SELECT PORT A
(2) 034772 013737 001224 001234      MOV  PORTA,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
(1)                                     ;*****
(1)                                     ;VERIFY THAT EITHER PORT A OR PORT B HAS THE DRIVE
(1) 035000 005760 000012                    TST  RPDS1(RO)      ;SEE THE REGISTER THROUGH PORT A ?
(1) 035004 001014                    BNE  1$            ;BR IF YES
(2) 035006 113760 001226 000010      MOVB PORTB,RPCS2(RO) ;SELECT PORT B
(2) 035014 013737 001226 001234      MOV  PORTB,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
(1) 035022 005760 000012                    TST  RPDS1(RO)      ;SEE REGISTER THROUGH PORT B
(1) 035026 001021                    BNE  2$            ;BR IF YES
(1) 035030 104042                    ERROR 42          ;DRIVE NOT IN NEUTRAL OR SEIZED
(1) 035032 000137 036602                    JMP  4$            ;BYPASS REST OF TEST
(1) 035036                                     1$:
(2) 035036 113760 001226 000010      MOVB PORTB,RPCS2(RO) ;SELECT PORT B
(2) 035044 013737 001226 001234      MOV  PORTB,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
(1) 035052 005760 000012                    TST  RPDS1(RO)      ;REGISTER SHOULD BE ZERO THROUGH PORT B
(1) 035056 001002                    BNE  +6           ;BR IF STATUS REG IS NOT ZERO
(1) 035060 000137 035732                    JMP  3$            ;STATUS REG IS ZERO
(1) 035064 104043                    ERROR 43          ;DRIVE IN NEUTRAL AFTER WRITE ATTN BIT
(1) 035066 000137 036602                    JMP  4$            ;BYPASS REST OF TEST
(1)                                     ;*****
(1)                                     ;PORT B HAS THE DRIVE. VERIFY THAT PORT A HAS PORT REQUEST SET
(1) 035072                                     2$:
(2) 035072 005037 001244                    CLR  CKERR          ;CLEAR THE 'CHECK ERROR' INDICATOR
(2) 035076 016037 000012 001126      MOV  RPDS1(RO), $BDDAT ;GET CONTENTS OF RPDS1
(2) 035104 012737 000012 001122      MOV  #RPDS1,$BDAOR  ;FORM REGISTER ADDRESS OF ERROR MESSAGE
(2) 035112 060037 001122                    ADD  RO,$BDAOR      ;ADD RHI1 BASE ADDRESS
(2) 035116 012737 011700 001124      MOV  #MOL!PGM!DPR!DRY!VV,$GDDAT ;WHAT REGISTER SHOULD BE
(2) 035124 013737 001126 001164      MOV  $BDDAT,$TMP0  ;MOVE REGISTER CONTENTS TO '$TMP0'
(2) 035132 042737 106077 001164      BIC  #1C71700,$TMP0 ;SAVE SPECIFIED BITS
(2) 035140 023737 001124 001164      CMP  $GDDAT,$TMP0  ;COMPARE THE BITS
(2) 035146 001414                    BEQ  64$          ;BR IF OK
(2) 035150 013737 001126 001174      MOV  $BDDAT,$TMP4  ;COPY 'BAD DATA'
(2) 035156 042737 071700 001174      BIC  #71700,$TMP4  ;CLEAR THE MASKED BITS
(2) 035164 053737 001174 001124      BIS  $TMP4,$GDDAT  ;'OR' WITH GOOD DATA FOR TYPEOUT
(2) 035172 104010                    ERROR 10          ;REPORT THE ERROR
(2) 035174 005137 001244                    COM  CKERR         ;SET THE REGISTER COMPARE ERROR INDICATOR
(1) 035200 000240                                     64$:
(1) 035202 013737 001226 001236      MOV  PORTB,SEIZPT ;ADDRESS FOR ERROR MESSAGE

```

```

(1) 035210 013737 001224 001240      MOV      PORTA,OPPRT      ;SAME AS ABOVE
(2)                                     ;RELEASE THE DRIVE FROM PORT B
(3) 035216 113760 001226 000010      MOV      PORTB,RPCS2(RO)  ;SELECT PORT B
(4) 035224 013737 001226 001234      MOV      PORTB,PTNBR     ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
(5) 035232 012760 000013 000000      MOV      #13,RPCS1(RO)   ;ISSUE RELEASE THROUGH PORT B
(6)                                     ;VERIFY THAT DRIVE IS SEIZED BY PORT A WHEN RELEASED BY PORT B
(7) 035240 005037 001250                CLR      RELERR          ;CLEAR 'RELEASE ERROR' INDICATOR
(8) 035244 012737 111700 001124      MOV      #ATA!MOL!PGM!DPR!DRY!VV,$GDDAT ;COMPARISON CONSTANT
(9) 035252 012737 000012 001122      MOV      #RPS1,$BADR     ;REGISTER ADDRESS INCREMENT
(10) 035260 060037 001122 001122      ADD      RO,$BADR        ;REGISTER BASE ADDRESS FOR TYPEOUT
(11) 035264 113760 001224 000010      MOV      PORTA,RPCS2(RO) ;SELECT PORT A
(12) 035272 013737 001224 001234      MOV      PORTA,PTNBR     ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
(13) 035300 016037 000012 001164      MOV      RPS1(RO),STMP0  ;READ STATUS REGISTER FROM PORT A
(14) 035306 113760 001226 000010      MOV      PORTB,RPCS2(RO) ;SELECT PORT B
(15) 035314 013737 001226 001234      MOV      PORTB,PTNBR     ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
(16) 035322 016037 000012 001126      MOV      RPS1(RO),$BDDAT ;DRIVE STATUS FROM PORT B
(17) 035330 001404                BEQ      66$             ;BR IF STATUS FROM PORT B ZERO
(18) 035332 005737 001164                TST      STMP0           ;IS STATUS FROM PORT A ZERO ?
(19) 035336 001401                BEQ      66$             ;BR IF ZERO
(20) 035340 104044                ERROR    44             ;REPORT DRIVE NOT SEIZED BY PORT A
(21) 035342 013737 001164 001126      MOV      STMP0,$BDDAT    ;CHECK STATUS FROM PORT A
(22) 035350 013737 001224 001234      MOV      PORTA,PTNBR     ;CHANGE PORT ADDRESS FOR TYPEOUT
(23) 035356 023737 001124 001126      CMP      $GDDAT,$BDDAT  ;COMPARE WITH CONSTANT
(24) 035364 001401                BEQ      67$             ;BR IF OK
(25) 035366 104027                ERROR    27             ;REPORT REGISTER ERROR
(26) 035370 000240                NOP
(27)                                     ;RELEASE THE DRIVE FROM PORT A
(28) 035372 113760 001224 000010      MOV      PORTA,RPCS2(RO) ;SELECT PORT A
(29) 035400 013737 001224 001234      MOV      PORTA,PTNBR     ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
(30) 035406 012760 000013 000000      MOV      #13,RPCS1(RO)  ;ISSUE RELEASE THROUGH PORT A
(31)                                     ;VERIFY THAT THE DRIVE IS IN NEUTRAL
(32) 035414 005037 001250                CLR      RELERR          ;CLEAR THE 'RELEASE ERROR' INDICATOR
(33) 035420 012737 000012 001122      MOV      #RPS1,$BADR     ;FORM THE ADDRESS OF RPS1 FOR TYPEOUT
(34) 035426 060037 001122 001122      ADD      RO,$BADR        ;ADD THE I/O BASE ADDRESS
(35) 035432 012737 011700 001124      MOV      #MOL!PGM!DPR!DRY!VV,$GDDAT    ;COMPARISON CONSTANT
(36) 035440 113760 001224 000010      MOV      PORTA,RPCS2(RO) ;SELECT PORT A.
(37) 035446 016037 000012 001170      MOV      RPS1(RO),STMP2 ;GET THE DRIVE STATUS REGISTER FROM PORT A.
(38) 035454 013737 001170 001164      MOV      STMP2,STMP0     ;COPY IT INTO 'STMP0'
(39) 035462 042737 100100 001164      BIC      #ATA!VV,STMP0   ;CLEAR PORT DEPENDENT BITS FROM THE COPY
(40) 035470 113760 001226 000010      MOV      PORTB,RPCS2(RO) ;SELECT PORT B.
(41) 035476 016037 000012 001172      MOV      RPS1(RO),STMP3 ;GET THE DRIVE STATUS REGISTER FROM PORT B.
(42) 035504 013737 001172 001166      MOV      STMP3,STMP1     ;COPY IT INTO 'STMP1'
(43) 035512 042737 100100 001166      BIC      #ATA!VV,STMP1   ;CLEAR PORT DEPENDENT BITS FROM THE COPY
(44) 035520 023737 001164 001166      CMP      STMP0,STMP1     ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
(45) 035526 001006                BNE      68$             ;BR IF NOT
(46) 035530 005737 001164                TST      STMP0           ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
(47) 035534 001045                BNE      70$             ;BR IF NOT
(48) 035536 104046                ERROR    46             ;REPORT DRIVE NOT IN NEUTRAL OR NOT SEIZED
    
```

```

(3) 035540 000137 035724          JMP          72$          ;BYPASS THE REST OF THE CHECKS
(3) 035544 013737 001170 001126 68$: MOV          $BDDAT      ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
(3) 035552 013737 001226 001234   MOV          PORTB,PTNBR ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
(3) 035560 113760 001226 000010   MOVVB       PORTB,RPCS2(RO) ;SELECT PORT B.
(3) 035566 005737 001164          TST          $TMP0       ;SEE IF STATUS EQ 0 FROM PORT A.
(3) 035572 001414          BEQ          69$        ;BR IF ZERO
(3) 035574 013737 001224 001234   MOV          PORTA,PTNBR ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
(3) 035602 013737 001172 001126   MOV          $TMP3,$BDDAT ;'BAD DATA' FOR ERROR TYPE OUT
(3) 035610 113760 001224 000010   MOVVB       PORTA,RPCS2(RO) ;SELECT PORT A.
(3) 035616 005737 001166          TST          $TMP1       ;SEE IF STATUS EQ ZERO FROM PORT B.
(3) 035622 001012          BNE          70$        ;BR IF NOT
(3) 035624 012737 177777 001250 69$: MOV          #-1,RELEA ;SET 'RELEASE ERROR' INDICATOR
(3) 035632 012760 000011 000000   MOV          #11,RPCS1(RO) ;CLEAR THE DRIVE
(3) 035640 012760 000013 000000   MOV          #13,RPCS1(RO) ;RELEASE THE DRIVE
(3) 035646 104026          ERROR       26          ;TYPE ERROR MESSAGE 26
(3) 035650 013737 001170 001126 70$: MOV          $TMP2,$BDDAT ;LOOK FOR BIT FAILURES W/IN RPDS1 READ
(3) 035656 013737 001224 001234   MOV          PORTA,PTNBR ;CHANGE PORT NUMBER
(3) 035664 023737 001124 001170   CMP          $GDDAT,$TMP2 ;ALL BITS OK?
(3) 035672 001401          BEQ          71$        ;BR IF OK FROM PORT A.
(3) 035674 104007          ERROR       7          ;REPORT ERROR
(3) 035676 013737 001172 001126 71$: MOV          $TMP3,$BDDAT ;CHECK RPDS1 FOR BIT FAILURES - FROM PORT B.
(3) 035704 013737 001226 001234   MOV          PORTB,PTNBR ;CHANGE PORT NUMBER
(3) 035712 023737 001124 001172   CMP          $GDDAT,$TMP3 ;SEE IF READ OK FROM PORT B.
(3) 035720 001401          BEQ          72$        ;BR IF OK
(3) 035722 104007          ERROR       7          ;REPORT ERROR
(3) 035724 000240          NOP
(3) 035726 000137 036602          JMP          4$

```

```

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

```

```

(1) 035732          3$: MOVVB       PORTA,RPCS2(RO) ;SELECT PORT A
(2) 035732 113760 001224 000010   MOV          PORTA,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
(2) 035740 013737 001224 001234   CLR          CKERR       ;CLEAR THE 'CHECK ERROR' INDICATOR
(2) 035746 005037 001244          MOV          RPDS1(RO), $BDDAT ;GET CONTENTS OF RPDS1
(2) 035752 016037 000012 001126   MOV          #RPDS1,$BDDADR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
(2) 035760 012737 000012 001122   ADD          R0,$BDDADR  ;ADD RH11 BASE ADDRESS
(2) 035766 060037 001122          MOV          #MOL!PGM!DPR!DRY!VV,$GDDAT ;WHAT REGISTER SHOULD BE
(2) 035772 012737 011700 001124   MOV          $BDDAT,$TMP0 ;MOVE REGISTER CONTENTS TO '$TMP0'
(2) 036000 013737 001126 001164   BIC          #1C71700,$TMP0 ;SAVE SPECIFIED BITS
(2) 036006 042737 106077 001164   CMP          $GDDAT,$TMP0 ;COMPARE THE BITS
(2) 036014 023737 001124 001164   BEQ          73$        ;BR IF OK
(2) 036022 001414          MOV          $BDDAT,$TMP4 ;COPY 'BAD DATA'
(2) 036024 013737 001126 001174   BIC          #71700,$TMP4 ;CLEAR THE MASKED BITS
(2) 036032 042737 071700 001174   BIS          $TMP4,$GDDAT ;'OR' WITH GOOD DATA FOR TYPEOUT
(2) 036040 053737 001174 001124   ERROR       10          ;REPORT THE ERROR
(2) 036046 104010          COM          CKERR       ;SET THE REGISTER COMPARE ERROR INDICATOR
(2) 036050 005137 001244          NOP
(2) 036054 000240          MOV          PORTA,SEIZPT ;ADDRESS FOR ERROR MESSAGE
(2) 036056 013737 001224 001236   MOV          PORTB,OPPRT ;SAME AS ABOVE
(2) 036064 013737 001226 001240          ;RELEASE THE DRIVE FROM PORT A

(2) 036072 113760 001224 000010   MOVVB       PORTA,RPCS2(RO) ;SELECT PORT A
(2) 036100 013737 001224 001234   MOV          PORTA,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT

```



```

(2) 036106 012760 000013 000000      MOV      #13,RPCS1(RO) ;ISSUE RELEASE THROUGH PORT A
(3)                                     ;VERIFY THAT DRIVE IS SEIZED BY PORT B WHEN RELEASED BY PORT A
(3) 036114 005037 001250                CLR      RELERR ;CLEAR 'RELEASE ERROR' INDICATOR
(3) 036120 012737 111700 001124      MOV      #ATA!MOL!PGM!DPR!DRY!VV,$GDDAT ;COMPARISON CONSTANT
(3) 036126 012737 000012 001122      MOV      #RPS1,$BDDADR ;REGISTER ADDRESS INCREMENT
(3) 036134 060037 001122                ADD      RO,$BDDADR ;REGISTER BASE ADDRESS FOR TYPEOUT
(4) 036140 113760 001226 000010      MOV      PORTB,RPCS2(RO) ;SELECT PORT B
(4) 036146 013737 001226 001234      MOV      PORTB,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
(4) 036154 016037 000012 001164      MOV      RPDS1(RO),$TMP0 ;READ STATUS REGISTER FROM PORT B
(4) 036162 113760 001224 000010      MOV      PORTA,RPCS2(RO) ;SELECT PORT A
(4) 036170 013737 001224 001234      MOV      PORTA,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
(3) 036176 016037 000012 001126      MOV      RPDS1(RO),$BDDAT ;DRIVE STATUS FROM PORT A
(3) 036204 001404                        BEQ      75$ ;BR IF STATUS FROM PORT A ZERO
(3) 036206 005737 001164                TST      $TMP0 ;IS STATUS FROM PORT B ZERO?
(3) 036212 001401                        BEQ      75$ ;BR IF ZERO
(3) 036214 104044                        ERROR    44 ;REPORT DRIVE NOT SEIZED BY PORT B
(3) 036216 013737 001164 001126 75$: MOV      $TMP0,$BDDAT ;CHECK STATUS FROM PORT B
(3) 036224 013737 001226 001234      MOV      PORTB,PTNBR ;CHANGE PORT ADDRESS FOR TYPEOUT
(3) 036232 023737 001124 001126      CMP      $GDDAT,$BDDAT ;COMPARE WITH CONSTANT
(3) 036240 001401                        BEQ      76$ ;BR IF OK
(3) 036242 104027                        ERROR    27 ;REPORT REGISTER ERROR
(3) 036244 000240                        NOP

(2)                                     ;RELEASE THE DRIVE FROM PORT B
(3) 036246 113760 001226 000010      MOV      PORTB,RPCS2(RO) ;SELECT PORT B
(3) 036254 013737 001226 001234      MOV      PORTB,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
(2) 036262 012760 000013 000000      MOV      #13,RPCS1(RO) ;ISSUE RELEASE THROUGH PORT B
(3)                                     ;VERIFY THAT THE DRIVE IS IN NEUTRAL
(3) 036270 005037 001250                CLR      RELERR ;CLEAR THE 'RELEASE ERROR' INDICATOR
(3) 036274 012737 000012 001122      MOV      #RPDS1,$BDDADR ;FORM THE ADDRESS OF RPDS1 FOR TYPEOUT
(3) 036302 060037 001122                ADD      RO,$BDDADR ;ADD THE I/O BASE ADDRESS
(3) 036306 012737 011700 001124      MOV      #MOL!PGM!DPR!DRY!VV,$GDDAT ;COMPARISON CONSTANT
(3) 036314 113760 001224 000010      MOV      PORTA,RPCS2(RO) ;SELECT PORT A.
(3) 036322 016037 000012 001170      MOV      RPDS1(RO),$TMP2 ;GET THE DRIVE STATUS REGISTER FROM PORT A.
(3) 036330 013737 001170 001164      MOV      $TMP2,$TMP0 ;COPY IT INTO '$TMP0'
(3) 036336 042737 100100 001164      BIC      #ATA!VV,$TMP0 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
(3) 036344 113760 001226 000010      MOV      PORTB,RPCS2(RO) ;SELECT PORT B.
(3) 036352 016037 000012 001172      MOV      RPDS1(RO),$TMP3 ;GET THE DRIVE STATUS REGISTER FROM PORT B.
(3) 036360 013737 001172 001166      MOV      $TMP3,$TMP1 ;COPY IT INTO '$TMP1'
(3) 036366 042737 100100 001166      BIC      #ATA!VV,$TMP1 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
(3) 036374 023737 001164 001166      CMP      $TMP0,$TMP1 ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS?
(3) 036402 001006                        BNE      77$ ;BR IF NOT
(3) 036404 005737 001164                TST      $TMP0 ;REGISTERS ARE THE SAME: ARE THEY ZERO?
(3) 036410 001045                        BNE      79$ ;BR IF NOT
(3) 036412 104046                        ERROR    46 ;REPORT DRIVE NOT IN NEUTRAL OR NOT SEIZED
(3) 036414 000137 036600                JMP      81$ ;BYPASS THE REST OF THE CHECKS
(3) 036420 013737 001170 001126 77$: MOV      $TMP2,$BDDAT ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
(3) 036426 013737 001226 001234      MOV      PORTB,PTNBR ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
(3) 036434 113760 001226 000010      MOV      PORTB,RPCS2(RO) ;SELECT PORT B.
(3) 036442 005737 001164                TST      $TMP0 ;SEE IF STATUS EQ 0 FROM PORT A.
(3) 036446 001414                        BEQ      78$ ;BR IF ZERO

```

(3)	036450	013737	001224	001234		MOV	PORTA,PTNBR	;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTPAL
(3)	036456	013737	001172	001126		MOV	\$TMP3,\$BDDAT	; 'BAD DATA' FOR ERROR TYPE OUT
(3)	036464	113760	001224	000010		MOVB	PORTA,RPCS2(RO)	;SELECT PORT A.
(3)	036472	005737	001166			TST	\$TMP1	;SEE IF STATUS EQ ZERO FROM PORT B.
(3)	036476	001012				BNE	79\$;BR IF NOT
(3)	036500	012737	177777	001250	78\$:	MOV	#-1,RELERR	;SET 'RELEASE ERROR' INDICATOR
(3)	036506	012760	000011	000000		MOV	#11,RPCS1(RO)	;CLEAR THE DRIVE
(3)	036514	012760	000013	000000		MOV	#13,RPCS1(RO)	;RELEASE THE DRIVE
(3)	036522	104026				ERROR	26	;TYPE ERROR MESSAGE 26
(3)	036524	013737	001170	001126	79\$:	MOV	\$TMP2,\$BDDAT	;LOOK FOR BIT FAILURES WHEN RPDS1 PEAC
(3)	036532	013737	001224	001234		MOV	PORTA,PTNBR	;CHANGE PORT NUMBER
(3)	036540	023737	001124	001170		CMP	\$GDDAT,\$TMP2	;ALL BITS OK ?
(3)	036546	001401				BEQ	80\$;BR IF OK FROM PORT A.
(3)	036550	104007				ERROR	7	;REPORT ERROR
(3)	036552	013737	001172	001126	80\$:	MOV	\$TMP3,\$BDDAT	;CHECK RPDS1 FOR BIT FAILURES - FROM PORT B.
(3)	036560	013737	001226	001234		MOV	PORTB,PTNBR	;CHANGE PORT NUMBER
(3)	036566	023737	001124	001172		CMP	\$GDDAT,\$TMP3	;SEE IF READ OK FROM PORT B.
(3)	036574	001401				BEQ	81\$;BR IF OK
(3)	036576	104007				ERROR	7	;REPORT ERROR
(3)	036600	000240			81\$:	NOP		
(1)	036602	000004			4\$:	SCOPE		;LOOP ?

8833
8845
8846
(3)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
8847
8865
(2)
(2)
(2)
(2)
(2)
(2)
(2)
(2)

```

*****
*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.
*
*****

```

(3)	036604					ST30:		
(3)	036604	005737	001274			TST	KYBCTL	;PERFORMING ONLY SINGLE TESTS ?
(3)	036610	001406				BEQ	2\$;BR IF NOT
(3)	036612	100002				BPL	1\$;BR IF JUST ENTERED TEST
(3)	036614	000137	002622			JMP	EXEC	;RETURN & GET NEXT TEST NUMBER
(3)	036620	012737	177777	001274	1\$:	MOV	#-1,KYBCTL	;SET SINGLE TEST INDICATOR
(3)	036626	112737	000030	001102	2\$:	MOVB	#30,\$STNM	;TEST NUMBER
(3)	036634	012737	036656	001106		MOV	#TEST30,\$LPADR	;LOAD LOOP ON TEST ADDRESS
(3)	036642	012737	036656	001110		MOV	#TEST30,\$LPERR	;LOAD LOOP ON ERROR ADDRESS
(1)	036650	012737	007640	001176		MOV	#4000,\$TIMES	;DO 4000. ITERATIONS
8847	036656	012706	001100		TEST30:	MOV	#STACK,SP	;LOAD THE STACK PINTER

;CLEAR ATTENTION BITS FOR BOTH PORTS

(2)	036662	113760	001224	000010		MOVB	PORTA,RPCS2(RO)	;SELECT PORT #A
(2)	036670	005060	000012			CLR	RPDS1(RO)	;SEIZE THE DRIVE
(2)	036674	012760	000011	000000		MOV	#11,RPCS1(RO)	;ISSUE DRIVE CLEAR
(2)	036702	012760	000013	000000		MOV	#13,RPCS1(RO)	;RELEASE THE DRIVE
(2)	036710	113760	001226	000010		MOVB	PORTB,RPCS2(RO)	;SELECT PORT #B
(2)	036716	005060	000012			CLR	RPDS1(RO)	;SEIZE THE DRIVE THROUGH PORT 'B'
(2)	036722	012760	000011	000000		MOV	#11,RPCS1(RO)	;ISSUE DRIVE CLEAR

```

(2) 036730 012760 000013 000000      MOV      #13,RPCS1(RO) ;RELEASE THE DRIVE
(1) 036736 113760 001230 000010      MOV      PORTC,RPCS2(RO) ;SELECT DRIVE NOT BEING TESTED
(1)
(2)
(1)
(1)
(1) 036744 013737 001232 001164      MOV      ASR,$STMP0 ;STORE ATTN BIT FOR PORT A
(1) 036752 005137 001164 001164      COM      $STMP0 ;COMPLEMENT IT
(1) 036756 013760 001164 000016      MOV      $STMP0,RPAS(RO) ;WRITE THE ATTN REGISTER
(1)
(2)
(1)
(1)
(2)
(2)
(2)
(2)
(2) 036764 005037 001250 001122      CLR      RELERR ;CLEAR THE 'RELEASE ERROR' INDICATOR
(2) 036770 012737 000012 001122      MOV      #RPDS1,$BDDADR ;FORM THE ADDRESS OF RPDS1 FOR TYPEOUT
(2) 036776 060037 001122 001122      ADD      RO,$BDDADR ;ADD THE I/O BASE ADDRESS
(2) 037002 012737 011700 001124      MOV      #MOL!PGM!DPR!DRY!VV,$GDDAT ;COMPARISON CONSTANT
(2) 037010 113760 001224 000010      MOV      PORTA,RPCS2(RO) ;SELECT PORT A.
(2) 037016 016037 000012 001170      MOV      RPDS1(RO),$STMP2 ;GET THE DRIVE STATUS REGISTER FROM PORT A.
(2) 037024 013737 001170 001164      MOV      $STMP2,$STMP0 ;COPY IT INTO '$STMP0'
(2) 037032 042737 100100 001164      BIC      #ATA!VV,$STMP0 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
(2) 037040 113760 001226 000010      MOV      PORTB,RPCS2(RO) ;SELECT PORT B.
(2) 037046 016037 000012 001172      MOV      RPDS1(RO),$STMP3 ;GET THE DRIVE STATUS REGISTER FROM PORT B.
(2) 037054 013737 001172 001166      MOV      $STMP3,$STMP1 ;COPY IT INTO '$STMP1'
(2) 037062 042737 100100 001166      BIC      #ATA!VV,$STMP1 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
(2) 037070 023737 001164 001166      CMP      $STMP0,$STMP1 ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
(2) 037076 001006 001164 001166      BNE      64$ ;BR IF NOT
(2) 037100 005737 001164 001166      TST      $STMP0 ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
(2) 037104 001045 001164 001166      BNE      66$ ;BR IF NOT
(2) 037106 104046 001164 001166      ERROR    46 ;REPORT DRIVE NOT IN NEUTRAL OR NOT SEIZED
(2) 037110 000137 037274 001126      JMP      68$ ;BYPASS THE REST OF THE CHECKS
(2) 037114 013737 001170 001126 64$: MOV      $STMP2,$GDDAT ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
(2) 037122 013737 001226 001234      MOV      PORTB,PTNBR ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
(2) 037130 113760 001226 000010      MOV      PORTB,RPCS2(RO) ;SELECT PORT B.
(2) 037136 005737 001164 001166      TST      $STMP0 ;SEE IF STATUS EQ 0 FROM PORT A.
(2) 037142 001414 001164 001166      BEQ      65$ ;BR IF ZERO
(2) 037144 013737 001224 001234      MOV      PORTA,PTNBR ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
(2) 037152 013737 001172 001126      MOV      $STMP3,$BDDAT ;'BAD DATA' FOR ERROR TYPE OUT
(2) 037160 113760 001224 000010      MOV      PORTA,RPCS2(RO) ;SELECT PORT A.
(2) 037166 005737 001166 001166      TST      $STMP1 ;SEE IF STATUS EQ ZERO FROM PORT B.
(2) 037172 001012 001166 001166      BNE      66$ ;BR IF NOT
(2) 037174 012737 177777 001250 65$: MOV      #-1,RELERR ;SET 'RELEASE ERROR' INDICATOR
(2) 037202 012760 000011 000000      MOV      #11,RPCS1(RO) ;CLEAR THE DRIVE
(2) 037210 012760 000013 000000      MOV      #13,RPCS1(RO) ;RELEASE THE DRIVE
(2) 037216 104021 001166 001166      ERROR    21 ;TYPE ERROR MESSAGE 21
(2) 037220 013737 001170 001126 66$: MOV      $STMP2,$BDDAT ;LOOK FOR BIT FAILURES WHEN RPDS1 READ
(2) 037226 013737 001224 001234      MOV      PORTA,PTNBR ;CHANGE PORT NUMBER
(2) 037234 023737 001124 001170      CMP      $GDDAT,$STMP2 ;ALL BITS OK ?
(2) 037242 001401 001166 001166      BEQ      67$ ;BR IF OK FROM PORT A.
(2) 037244 104007 001166 001166      ERROR    7 ;REPORT ERROR
(2) 037246 013737 001172 001126 67$: MOV      $STMP3,$BDDAT ;CHECK RPDS1 FOR BIT FAILURES - FROM PORT B.
(2) 037254 013737 001226 001234      MOV      PORTB,PTNBR ;CHANGE PORT NUMBER
(2) 037262 023737 001124 001172      CMP      $GDDAT,$STMP3 ;SEE IF READ OK FROM PORT B.

```

(2) 037270 001401
 (2) 037272 104007
 (2) 037274 000240
 (1) 037276 000004
 8866
 8880
 8881
 (2) 037300
 (3) 037300 005737 001274
 (3) 037304 001406
 (3) 037306 100002
 (3) 037310 000137 002622
 (3) 037314 012737 177777 001274
 (3) 037322 112737 000031 001102
 (3) 037330 012737 037352 001106
 (3) 037336 012737 037352 001110
 (1) 037344 012737 000004 001176
 8882 037352 012706 001100
 8938
 (2)
 (2) 037356 113760 001224 000010
 (2) 037364 005060 000012 000000
 (2) 037370 012760 000011 000000
 (2) 037376 012760 000013 000000
 (2) 037404 113760 001226 000010
 (2) 037412 005060 000012 000000
 (2) 037416 012760 000011 000000
 (2) 037424 012760 000013 000000
 (2)
 (2)
 (2) 037432 113760 001224 000010
 (2) 037440 013737 001224 001236
 (2) 037446 005060 000012 000000
 (2) 037452 013737 001226 001240
 (1)
 (2)
 (1)
 (1) 037460 012760 177777 000014
 (2)

```

      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.
*
:*****
†ST31:
      TST     KYBCTL    ;PERFORMING ONLY SINGLE TESTS ?
      BEQ     2$        ;BR IF NOT
      BPL     1$        ;BR IF JUST ENTERED TEST
      JMP     EXEC      ;RETURN & GET NEXT TEST NUMBER
1$:   MOV     #-1,KYBCTL ;SET SINGLE TEST INDICATOR
2$:   MOVB    #31,$STNM ;TEST NUMBER
      MOV     #TEST31,$LPADR ;LOAD LOOP ON TEST ADDRESS
      MOV     #TEST31,$LPERR ;LOAD LOOP ON ERROR ADDRESS
      MOV     #4,$TIMES  ;DO 4 ITERATIONS
TES*31: MOV     #STACK,$P ;LOAD THE STACK POINTER

      ;CLEAR ATTENTION BITS FOR BOTH PORTS

      MOVB   PORTA,RPDS2(RO) ;SELECT PORT #A
      CLR    RPDS1(RO)       ;SEIZE THE DRIVE
      MOV    #11,RPDS1(RO)   ;ISSUE DRIVE CLEAR
      MOV    #13,RPDS1(RO)   ;RELEASE THE DRIVE
      MOVB  PORTB,RPDS2(RO) ;SELECT PORT #B
      CLR    RPDS1(RO)       ;SEIZE THE DRIVE THROUGH PORT 'B'
      MOV    #11,RPDS1(RO)   ;ISSUE DRIVE CLEAR
      MOV    #13,RPDS1(RO)   ;RELEASE THE DRIVE
;*****

      ;SEIZE THE DRIVE THROUGH PORT A

      MOVB   PORTA,RPDS2(RO) ;SELECT PORT A
      MOV    PORTA,SEIZPT    ;STORE SEIZING PORT'S ADDRESS
      CLR    RPDS1(RO)       ;WRITE RPDS1
      MOV    PORTB,OPPR1     ;'OPPOSITE' PORT ADDRESS

:*****
;FORCE AN ERROR
      MOV    #-1,RPER1(RO)  ;SET ERROR BITS

```

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

037466 005037 001252
037472 012737 003720 001254
037500 113760 001226 000010
037506 013737 001226 001234

037514 005760 000012
037520 001004
037522 005737 001254
037526 001372
037530 104036
037532
037532 113760 001224 000010
037540 013737 001224 001234

037546 005037 001244
037552 016037 000012 001126
037560 012737 000012 001122
037566 060037 001122
037572 012737 040000 001124
037600 013737 001126 001164
037606 042737 137777 001164
037614 023737 001124 001164
037622 001414
037624 013737 001126 001174
037632 042737 040000 001174
037640 053737 001174 001124
037646 104023
037650 005137 001244
037654 000240

037656 005037 001244
037662 016037 000014 001126
037670 012737 000014 001122
037676 060037 001122
037702 012737 177777 001124
037710 023737 001124 001126
037716 001403
037720 104010
037722 005137 001244
037726 000240

```
*****
;START THE TIMER
CLR TIME ;CLEAR THE ELAPSED TIME COUNTER
MOV #2000, WATCH ;SET WATCH TO 2000 MS
MOVB PORTB, RPCS2(RO) ;SELECT PORT B
MOV PORTB, PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT

*****
;WAIT FOR DRIVE TO TIMEOUT
1$: TST RPDS1(RO) ;WAIT FOR THE DRIVE TO BE RELEASED
    BNE 2$ ;BR IF DRIVE RELEASED
    TST WATCH ;WATCH AT ZERO ?
    BNE 1$ ;BR IF NOT
    ERROR 36 ;DRIVE NOT RELEASED WITHIN 2 SECONDS

2$: MOVB PORTA, RPCS2(RO) ;SELECT PORT A
    MOV PORTA, PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT

*****
;THE ERROR BIT ('ERR') IN RPDS1 SHOULD STILL BE SET
CLR CKERR ;CLEAR THE 'CHECK ERROR' INDICATOR
MOV RPDS1(RO), $BDDAT ;GET CONTENTS OF RPDS1
MOV #RPDS1, $BDAOR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
ADD RO, $BDAOR ;ADD RH11 BASE ADDRESS
MOV #ERR, $GDDAT ;WHAT REGISTER SHOULD BE
MOV $BDDAT, $TMP0 ;MOVE REGISTER CONTENTS TO '$TMP0'
BIC #140000, $TMP0 ;SAVE SPECIFIED BITS
CMP $GDDAT, $TMP0 ;COMPARE THE BITS
BEQ 66$ ;BR IF OK
MOV $BDDAT, $TMP4 ;COPY 'BAD DATA'
BIC #40000, $TMP4 ;CLEAR THE MASKED BITS
BIS $TMP4, $GDDAT ;'OR' WITH GOOD DATA FOR TYPEOUT
ERROR 23 ;TYPE MESSAGE 23
COM CKERR ;SET THE REGISTER COMPARE ERROR INDICATOR
66$: NOP

*****
;THE ERROR REGISTER SHOULD CONTAIN 1'S
CLR CKERR ;CLEAR THE 'CHECK ERROR' INDICATOR
MOV RPER1(RO), $BDDAT ;GET CONTENTS OF RPER1
MOV #RPER1, $BDAOR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
ADD RO, $BDAOR ;ADD RH11 BASE ADDRESS
MOV #177777, $GDDAT ;WHAT REGISTER SHOULD BE
CMP $GDDAT, $BDDAT ;IS THE REGISTER OK ?
BEQ 68$ ;BR IF OK
ERROR 10 ;REPORT THE ERROR
COM CKERR ;SET THE REGISTER COMPARE ERROR INDICATOR
68$: NOP

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

```

(2) 037730 005037 001244 CLR CKERR ;CLEAR THE 'CHECK ERROR' INDICATOR
(2) 037734 016037 000012 001126 MOV #RPS1,$BDDAT ;GET CONTENTS OF RPS1
(2) 037742 012737 000012 001122 MOV #RPS1,$BDAOR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
(2) 037750 060037 001122 ADD R0,$BDAOR ;ADD RHI1 BASE ADDRESS
(2) 037754 012737 100000 001124 MOV #ATA,$GDDAT ;WHAT REGISTER SHOULD BE
(2) 037762 013737 001126 001164 MOV $BDDAT,$TMP0 ;MOVE REGISTER CONTENTS TO '$TMP0'
(2) 037770 042737 077777 001164 BIC #1CHTA,$TMP0 ;SAVE SPECIFIED BITS
(2) 037776 023737 001124 001164 CMP $GDDAT,$TMP0 ;COMPARE THE BITS
(2) 040004 001414 70S: BEQ 70S ;BR IF OK
(2) 040006 013737 001126 001174 MOV $BDDAT,$TMP4 ;COPY 'BAD DATA'
(2) 040014 042737 100000 001174 BIC #ATA,$TMP4 ;CLEAR THE MASKED BITS
(2) 040022 053737 001174 001124 BIS $TMP4,$GDDAT ;'OR' WITH GOOD DATA FOR TYPEOUT
(2) 040030 104041 ERROR 41 ;TYPE MESSAGE 41
(2) 040032 005137 001244 COM CKERR ;SET THE REGISTER COMPARE ERROR INDICATOR
(2) 040036 000240 NOP

```

;*****

;VERIFY THAT THE DRIVE IS IN NEUTRAL

```

(2) 040040 005037 001250 CLR RELERR ;CLEAR THE 'RELEASE ERROR' INDICATOR
(2) 040044 012737 000012 001122 MOV #RPS1,$BDAOR ;FORM THE ADDRESS OF RPS1 FOR TYPEOUT
(2) 040052 060037 001122 ADD R0,$BDAOR ;ADD THE I/O BASE ADDRESS
(2) 040056 012737 051700 001124 MOV #51700,$GDDAT ;COMPARISON CONSTANT
(2) 040064 113760 001224 000010 MOVB PORTA,$PCS2(R0) ;SELECT PORT A.
(2) 040072 016037 000012 001170 MOV $RPS1(R0),$TMP2 ;GET THE DRIVE STATUS REGISTER FROM PORT A.
(2) 040100 013737 001170 001164 MOV $TMP2,$TMP0 ;COPY IT INTO '$TMP0'
(2) 040106 042737 100100 001164 BIC #ATA!VV,$TMP0 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
(2) 040114 113760 001226 000010 MOVB PORTB,$PCS2(R0) ;SELECT PORT B.
(2) 040122 016037 000012 001172 MOV $RPS1(R0),$TMP3 ;GET THE DRIVE STATUS REGISTER FROM PORT B.
(2) 040130 013737 001172 001166 MOV $TMP3,$TMP1 ;COPY IT INTO '$TMP1'
(2) 040136 042737 100100 001166 BIC #ATA!VV,$TMP1 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
(2) 040144 023737 001164 001166 CMP $TMP0,$TMP1 ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
(2) 040152 001006 72S: BNE 72S ;BR IF NOT
(2) 040154 005737 001164 TST $TMP0 ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
(2) 040160 001045 74S: BNE 74S ;BR IF NOT
(2) 040162 104046 ERROR 46 ;REPORT DRIVE NOT IN NEUTRAL OR NOT SEIZED
(2) 040164 000137 040364 JMP 76S ;BYPASS THE REST OF THE CHECKS
(2) 040170 013737 001170 001126 72S: MOV $TMP2,$BDDAT ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
(2) 040176 013737 001226 001234 MOV PORTB,$PTNBR ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
(2) 040204 113760 001226 000010 MOVB PORTB,$PCS2(R0) ;SELECT PORT B.
(2) 040212 005737 001164 TST $TMP0 ;SEE IF STATUS EQ 0 FROM PORT A.
(2) 040216 001414 73S: BEQ 73S ;BR IF ZERO
(2) 040220 013737 001224 001234 MOV PORTA,$PTNBR ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
(2) 040226 013737 001172 001126 MOV $TMP3,$BDDAT ;'BAD DATA' FOR ERROR TYPE OUT
(2) 040234 113760 001224 000010 MOVB PORTA,$PCS2(R0) ;SELECT PORT A.
(2) 040242 005737 001166 TST $TMP1 ;SEE IF STATUS EQ ZERO FROM PORT B.
(2) 040246 001012 74S: BNE 74S ;BR IF NOT
(2) 040250 012737 177777 001250 73S: MOV #-1,RELERR ;SET 'RELEASE ERROR' INDICATOR
(2) 040256 012760 000011 000000 MOV #11,$RPS1(R0) ;CLEAR THE DRIVE
(2) 040264 012760 000013 000000 MOV #13,$RPS1(R0) ;RELEASE THE DRIVE
(2) 040272 104026 ERROR 26 ;TYPE ERROR MESSAGE 26
(2) 040274 013737 001170 001126 74S: MOV $TMP2,$BDDAT ;LOOK FOR BIT FAILURES WHEN RPS1 READ
(2) 040302 013737 001224 001234 MOV PORTA,$PTNBR ;CHANGE PORT NUMBER
(2) 040310 042737 100000 001170 BIC #ATA,$TMP2 ;DON'T CHECK THE ATTN BIT

```

```

(2) 040316 023737 001124 001170      CMP      $GDDAT,$TMP2      ;ALL BITS OK ?
(2) 040324 001401                      BEQ      75$               ;BR IF OK FROM PORT A.
(2) 040326 104007                      ERROR   7                 ;REPORT ERROR
(2) 040330 013737 001172 001126 75$: MOV      $TMP3,$BDDAT      ;CHECK RPDS1 FOR BIT FAILURES - FROM PORT B.
(2) 040336 013737 001226 001234      MOV      PORTB,PTNBR      ;CHANGE PORT NUMBER
(2) 040344 042737 100000 001172      BIC     #ATA,$TMP3        ;DON'T CHECK THE ATTN BIT
(2) 040352 023737 001124 001172      CMP      $GDLAT,$TMP3     ;SEE IF READ OK FROM PORT B.
(2) 040360 001401                      BEQ      76$               ;BR IF OK
(2) 040362 104007                      ERROR   7                 ;REPORT ERROR
(2) 040364 000240                      NOP

```

```

:*****
;THE ATTENTION BIT FOR PORT B SHOULD NOT BE SET

```

```

(2) 040366 113760 001226 000010      MOVVB   PORTB,RPCS2(RO)   ;SELECT PORT B
(2) 040374 013737 001226 001234      MOV     PORTB,PTNBR      ;MOVE PORT ADDRESS TO LOCAT.JN FOR TYPEOUT
(2) 040402 005037 001244                      CKERR   ;CLEAR THE 'CHECK ERROR' INDICATOR
(2) 040406 016037 000012 001126      MOV     RPDS1(RO),$BDDAT  ;GET CONTENTS OF RPDS1
(2) 040414 012737 000012 001122      MOV     #RPDS1,$BADR      ;FORM REGISTER ADDRESS OF ERROR MESSAGE
(2) 040422 060037 001122                      ADD     RO,$BADR          ;ADD RHI1 BASE ADDRESS
(2) 040426 005037 001124                      CLR     $GDDAT           ;WHAT REGISTER SHOULD BE
(2) 040432 013737 001126 001164      MOV     $BDDAT,$TMP0     ;MOVE REGISTER CONTENTS TO '$TMP0'
(2) 040440 042737 077777 001164      BIC     #1CATA,$TMP0     ;SAVE SPECIFIED BITS
(2) 040446 023737 001124 001164      CMP     $GDDAT,$TMP0     ;COMPARE THE BITS
(2) 040454 001414                      BEQ     77$               ;BR IF OK
(2) 040456 013737 001126 001174      MOV     $BDDAT,$TMP4     ;COPY 'BAD DATA'
(2) 040464 042737 100000 001174      BIC     #ATA,$TMP4       ;CLEAR THE MASKED BITS
(2) 040472 053737 001174 001124      BIS     $TMP4,$GDDAT     ;'OR' WITH GOOD DATA FOR TYPEOUT
(2) 040500 104052                      ERROR   52                ;TYPE MESSAGE 52
(2) 040502 005137 001244                      COM     CKERR             ;SET THE REGISTER COMPARE ERROR INDICATOR
(2) 040506 000240                      NOP

```

77\$:

;CLEAR ATTENTION BIT FOR PORT A

```

(2) 040510 113760 001224 000010      MOVVB   PORTA,RPCS2(RO)   ;SELECT PORT #A
(2) 040516 005060 000012                      CLR     RPDS1(RO)        ;SEIZE THE DRIVE
(2) 040522 012760 000011 000000      MOV     #11,RPCS1(RO)    ;ISSUE DRIVE CLEAR
(2) 040530 012760 000013 000000      MOV     #13,RPCS1(RO)    ;RELEASE THE DRIVE
(2) 040536 000004                      SCOPE   ;LOOP ?

```

3\$:

```

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

```

890503

```

(3)
(4)
(4)
(4)
(4)
(4)
(4)
(4)
(4)
(4)
(3)
(3)

```

```

:*****
†ST32:      TST      KYBCTL      ;PERFORMING ONLY SINGLE TESTS ?

```

```

(3) 040544 001406 BEQ 2$ ;BR IF NOT
(3) 040546 100002 BPL 1$ ;BR IF JUST ENTERED TEST
(3) 040550 000137 002622 JMP EXEC ;RETURN & GET NEXT TEST NUMBER
(3) 040554 012737 177777 001274 1$: MOV #-1,KYBCTL ;SET SINGLE TEST INDICATOR
(3) 040562 112737 000032 001102 2$: MOVB #32,$STNM ;TEST NUMBER
(3) 040570 012737 040612 001106 MOV #TEST32,$LPADR ;LOAD LOOP ON TEST ADDRESS
(3) 040576 012737 040612 001110 MOV #TEST32,$LPERR ;LOAD LOOP ON ERROR ADDRESS
(1) 040604 012737 000004 001176 MOV #4,$TIMES ;DO 4 ITERATIONS
8954 040612 012706 001100 TEST32: MOV #STACK,$P ;LOAD THE STACK POINTER
8955
(2) ;CLEAR ATTENTION BITS FOR BOTH PORTS
(2) 040616 113760 001224 000010 MOVB PORTA,RPCS2(RO) ;SELECT PORT #A
(2) 040624 005060 000012 CLR RPS1(RO) ;SEIZE THE DRIVE
(2) 040630 012760 000011 000000 MOV #11,RPCS1(RO) ;ISSUE DRIVE CLEAR
(2) 040636 012760 000013 000000 MOV #13,RPCS1(RO) ;RELEASE THE DRIVE
(2) 040644 113760 001226 000010 MOVB PORTB,RPCS2(RO) ;SELECT PORT #B
(2) 040652 005060 000012 CLR RPS1(RO) ;SEIZE THE DRIVE THROUGH PORT 'B'
(2) 040656 012760 000011 000000 MOV #11,RPCS1(RO) ;ISSUE DRIVE CLEAR
(2) 040664 012760 000013 000000 MOV #13,RPCS1(RO) ;RELEASE THE DRIVE
;*****
(2) ;SEIZE THE DRIVE THROUGH PORT B
(2) 040672 113760 001226 000010 MOVB PORTB,RPCS2(RO) ;SELECT PORT B
(2) 040700 013737 001226 001236 MOV PORTB,SEIZPT ;STORE SEIZING PORT'S ADDRESS
(2) 040706 005060 000012 CLR RPS1(RO) ;WRITE RPS1
(2) 040712 013737 001224 001240 MOV PORTA,$PPRT ;'OPPOSITE' PORT ADDRESS
;*****
(2) ;FORCE AN ERROR
(1) 040720 012760 177777 000014 MOV #-1,RPER1(RO) ;SET ERROR BITS
;*****
(2) ;START THE TIMER
(2) 040726 005037 001252 CLR TIME ;CLEAR THE ELAPSED TIME COUNTER
(2) 040732 012737 003720 001254 MOV #2000,WATCH ;SET WATCH TO 2000 MS
(2) 040740 113760 001224 000010 MOVB PORTA,RPCS2(RO) ;SELECT PORT A
(2) 040746 013737 001224 001234 MOV PORTA,$PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
;*****
(1) ;WAIT FOR DRIVE TO TIMEOUT
(1) 040754 005760 000012 1$: TST RPS1(RO) ;WAIT FOR THE DRIVE TO BE RELEASED
(1) 040760 001004 BNE 2$ ;BR IF DRIVE RELEASED
(1) 040762 005737 001254 TST WATCH ;WATCH AT ZERO ?
(1) 040766 001372 BNE 1$ ;BR IF NOT
(1) 040770 104036 ERROR 36 ;DRIVE NOT RELEASED WITHIN 2 SECONDS
(2) 040772 113760 001226 000010 2$: MOVB PORTB,RPCS2(RO) ;SELECT PORT B
(2) 041000 013737 001226 001234 MOV PORTB,$PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
;*****
;THE ERROR BIT ('ERR') IN RPS1 SHOULD STILL BE SET

```



```

(1)
(R) 041006 005037 001244 CLR CKERR ;CLEAR THE 'CHECK ERROR' INDICATOR
(R) 041012 016037 000012 001126 MOV RPDS1(RC), $BDDAT ;GET CONTENTS OF RPDS1
(R) 041020 012737 000012 001122 MOV #RPDS1, $BDAOR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
(R) 041026 060037 001122 ADD RO, $BDAOR ;ADD RH11 BASE ADDRESS
(R) 041032 012737 040000 001124 MOV #ERR, $GDDAT ;WHAT REGISTER SHOULD BE
(R) 041040 013737 001126 001164 MOV $BDDAT, $TMP0 ;MOVE REGISTER CONTENTS TO '$TMP0'
(R) 041046 042737 137777 001164 BIC #1C4000, $TMP0 ;SAVE SPECIFIED BITS
(R) 041054 023737 001124 001164 CMP $GDDAT, $TMP0 ;COMPARE THE BITS
(R) 041062 001414 BEQ 66$ ;BR IF OK
(R) 041064 013737 001126 001174 MOV $BDDAT, $TMP4 ;COPY 'BAD DATA'
(R) 041072 042737 040000 001174 BIC #40000, $TMP4 ;CLEAR THE MASKED BITS
(R) 041100 053737 001174 001124 BIC $TMP4, $GDDAT ;'OR' WITH GOOD DATA FOR TYPEOUT
(R) 041106 104023 ERROR 23 ;TYPE MESSAGE 23
(R) 041110 005137 001244 COM CKERR ;SET THE REGISTER COMPARE ERROR INDICATOR
(R) 041114 000240 66$: NOP

```

;THE ERROR REGISTER SHOULD CONTAIN 1'S

```

(R) 041116 005037 001244 CLR CKERR ;CLEAR THE 'CHECK ERROR' INDICATOR
(R) 041122 016037 000014 001126 MOV RPER1(RO), $BDDAT ;GET CONTENTS OF RPER1
(R) 041130 012737 000014 001122 MOV #RPER1, $BDAOR ;FORM REGISTER ADDRESS OF ERPOR MESSAGE
(R) 041136 060037 001122 ADD RO, $BDAOR ;ADD RH11 BASE ADDRESS
(R) 041142 012737 177777 001124 MOV #177777, $GDDAT ;WHAT REGISTER SHOULD BE
(R) 041150 023737 001124 001126 CMP $GDDAT, $BDDAT ;IS THE REGISTER OK ?
(R) 041156 001403 BEQ 68$ ;BR IF OK
(R) 041160 104010 ERROR 10 ;REPORT THE ERROR
(R) 041162 005137 001244 COM CKERR ;SET THE REGISTER COMPARE ERROR INDICATOR
(R) 041166 000240 68$: NOP

```

;THE ATTENTION BIT FOR PORT B SHOULD STILL BE SET

```

(R) 041170 005037 001244 CLR CKERR ;CLEAR THE 'CHECK ERROR' INDICATOR
(R) 041174 016037 000012 001126 MOV RPDS1(RO), $BDDAT ;GET CONTENTS OF RPDS1
(R) 041202 012737 000012 001122 MOV #RPDS1, $BDAOR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
(R) 041210 060037 001122 ADD RO, $BDAOR ;ADD RH11 BASE ADDRESS
(R) 041214 012737 100000 001124 MOV #ATA, $GDDAT ;WHAT REGISTER SHOULD BE
(R) 041222 013737 001126 001164 MOV $BDDAT, $TMP0 ;MOVE REGISTER CONTENTS TO '$TMP0'
(R) 041230 042737 077777 001164 BIC #1CATA, $TMP0 ;SAVE SPECIFIED BITS
(R) 041236 023737 001124 001164 CMP $GDDAT, $TMP0 ;COMPARE THE BITS
(R) 041244 001414 BEQ 70$ ;BR IF OK
(R) 041246 013737 001126 001174 MOV $BDDAT, $TMP4 ;COPY 'BAD DATA'
(R) 041254 042737 100000 001174 BIC #ATA, $TMP4 ;CLEAR THE MASKED BITS
(R) 041262 053737 001174 001124 BIS $TMP4, $GDDAT ;'OR' WITH GOOD DATA FOR TYPEOUT
(R) 041270 104041 ERROR 41 ;TYPE MESSAGE 41
(R) 041272 005137 001244 COM CKERR ;SET THE REGISTER COMPARE ERROR INDICATOR
(R) 041276 000240 70$: NOP

```

;VERIFY THAT THE DRIVE IS IN NEUTRAL

```

(R) 041300 005037 001250 CLR RELERR ;CLEAR THE 'RELEASE ERROR' INDICATOR

```

E10

CZRJECO, DL CTRLR LGC MACY11 30A(1052) 28-DEC-77 10:17 PAGE 67-28
CZRJEC.P11 21-DEC-77 14:19

T32 TEST PORT 'B' TIMEOUT DOES NOT RESET DRIVE

SEG 0121

```

(2) 041304 012737 000012 001122 MOV #RPDS1,$BDADR ;FORM THE ADDRESS OF RPDS1 FOR TYPEOUT
(2) 041312 060037 001122 ADD RO,$BDADR ;ADD THE I/O BASE ADDRESS
(2) 041316 012737 051700 001124 MOV #51700,$GDDAT ;COMPARISON CONSTANT
(2) 041324 113760 001224 000010 MOVB PORTA,RPCS2(RO) ;SELECT PORT A.
(2) 041332 016037 000012 001170 MOV RPDS1(RO),$TMP2 ;GET THE DRIVE STATUS REGISTER FROM PORT A.
(2) 041340 013737 001170 001164 MOV $TMP2,$TMP0 ;COPY IT INTO '$TMP0'
(2) 041346 042737 100100 001164 BIC #ATA!VV,$TMP0 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
(2) 041354 113760 001226 000010 MOVB PORTB,RPCS2(RO) ;SELECT PORT B.
(2) 041362 016037 000012 001172 MOV RPDS1(RO),$TMP3 ;GET THE DRIVE STATUS REGISTER FROM PORT B.
(2) 041370 013737 001172 001166 MOV $TMP3,$TMP1 ;COPY IT INTO '$TMP1'
(2) 041376 042737 100100 001166 BIC #ATA!VV,$TMP1 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
(2) 041404 023737 001164 001166 CMP $TMP0,$TMP1 ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
(2) 041412 001006 BNE 72$ ;BR IF NOT
(2) 041414 005737 001164 TST $TMP0 ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
(2) 041420 001045 BNE 74$ ;BR IF NOT
(2) 041422 104046 ERROR 46 ;REPORT DRIVE NOT IN NEUTRAL OR NOT SEIZED
(2) 041424 000137 041624 JMP 76$ ;BYPASS THE REST OF THE CHECKS
(2) 041430 013737 001170 001126 72$: MOV $TMP2,$BDAT ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
(2) 041436 013737 001226 001234 MOV PORTB,PTNBR ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
(2) 041444 113760 001226 000010 MOVB PORTB,RPCS2(RO) ;SELECT PORT B.
(2) 041452 005737 001164 TST $TMP0 ;SEE IF STATUS EQ 0 FROM PORT A.
(2) 041456 001414 BEQ 73$ ;BR IF ZERO
(2) 041460 013737 001224 001234 MOV PORTA,PTNBR ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
(2) 041466 013737 001172 001126 MOV $TMP3,$BDAT ;'BAD DATA' FOR ERROR TYPE OUT
(2) 041474 113760 001224 000010 MOVB PORTA,RPCS2(RO) ;SELECT PORT A.
(2) 041502 005737 001166 TST $TMP1 ;SEE IF STATUS EQ ZERO FROM PORT B.
(2) 041506 001012 BNE 74$ ;BR IF NOT
(2) 041510 012737 177777 001250 73$: MOV #-1,RELERR ;SET 'RELEASE ERROR' INDICATOR
(2) 041516 012760 000011 000000 MOV #11,RPCS1(RO) ;CLEAR THE DRIVE
(2) 041524 012760 000013 000000 MOV #13,RPCS1(RO) ;RELEASE THE DRIVE
(2) 041532 104026 ERROR 26 ;TYPE ERROR MESSAGE 26
(2) 041534 013737 001170 001126 74$: MOV $TMP2,$BDAT ;LOOK FOR BIT FAILURES WHEN RPDS1 READ
(2) 041542 013737 001224 001234 MOV PORTA,PTNBR ;CHANGE PORT NUMBER
(2) 041550 042737 100000 001170 BIC #ATA,$TMP2 ;DON'T CHECK THE ATTN BIT
(2) 041556 023737 001124 001170 CMP $GDDAT,$TMP2 ;ALL BITS OK ?
(2) 041564 001401 BEQ 75$ ;BR IF OK FROM PORT A.
(2) 041566 104007 ERROR 7 ;REPORT ERROR
(2) 041570 013737 001172 001126 75$: MOV $TMP3,$BDAT ;CHECK RPDS1 FOR BIT FAILURES - FROM PORT B.
(2) 041576 013737 001226 001234 MOV PORTB,PTNBR ;CHANGE PORT NUMBER
(2) 041604 042737 100000 001172 BIC #ATA,$TMP3 ;DON'T CHECK THE ATTN BIT
(2) 041612 023737 001124 001172 CMP $GDDAT,$TMP3 ;SEE IF READ OK FROM PORT B.
(2) 041620 001401 BEQ 76$ ;BR IF OK
(2) 041622 104007 ERROR 7 ;REPORT ERROR
(2) 041624 000240 76$: NOP

```

;THE ATTENTION BIT FOR PORT A SHOULD NOT BE SET

```

(2) 041626 113760 001224 000010 MOVB PORTA,RPCS2(RO) ;SELECT PORT A
(2) 041634 013737 001224 001234 MOV PORTA,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
(2) 041642 005037 001244 CLR CKERR ;CLEAR THE 'CHECK ERROR' INDICATOR
(2) 041646 016037 000012 001126 MOV RPDS1(RO),$BDAT ;GET CONTENTS OF RPDS1
(2) 041654 012737 000012 001122 MOV #RPDS1,$BDADR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
(2) 041662 060037 001122 ADD RO,$BDADR ;ADD RH11 BASE ADDRESS
(2) 041666 005037 001124 CLR $GDDAT ;WHAT REGISTER SHOULD BE
(2) 041672 013737 001126 001164 MOV $BDAT,$TMP0 ;MOVE REGISTER CONTENTS TO '$TMP0'

```

```

(2) 041700 042737 077777 001164 BIC #1CATA,$TMPD ;SAVE SPECIFIED BITS
(2) 041706 023737 001124 001164 CMP $GDDAT,$TMPD ;COMPARE THE BITS
(2) 041714 001414 BEQ 77$ ;BR IF OK
(2) 041716 013737 001126 001174 MOV $BDDAT,$TMP4 ;COPY 'BAD DATA'
(2) 041724 042737 100000 001174 BIC #ATA,$TMP4 ;CLEAR THE MASKED BITS
(2) 041732 053737 001174 001124 BIS $TMP4,$GDDAT ;'OR' WITH GOOD DATA FOR TYPEOUT
(2) 041740 104052 ERROR 52 ;TYPE MESSAGE 52
(2) 041742 005137 001244 COM CKERR ;SET THE REGISTER COMPARE ERROR INDICATOR
(2) 041746 000240 NOP

```

;CLEAR ATTENTION BIT FOR PORT B

```

(2) 041750 113760 001226 000010 MOVB PORTB,RPCS2(RO) ;SELECT PORT #B
(2) 041756 005060 000012 CLR RPDS1(RO) ;SEIZE THE DRIVE
(2) 041762 012760 000011 000000 MOV #11,RPCS1(RO) ;ISSUE DRIVE CLEAR
(2) 041770 012760 000013 000000 MOV #13,RPCS1(RO) ;RELEASE THE DRIVE
(1) 041776 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.
*
*****

```

```

(2) 042000
(3) 042000 005737 001274 TST KYBCTL ;PERFORMING ONLY SINGLE TESTS
(3) 042004 001406 BEQ 2$ ;BR IF NOT
(3) 042006 100002 BPL 1$ ;BR IF JUST ENTERED TEST
(3) 042010 000137 002522 JMP EXEC ;RETURN & GET NEXT TEST NUMBER
(3) 042014 012737 177777 001274 1$: MOV #-1,KYBCTL ;SET SINGLE TEST INDICATOR
(3) 042022 112737 000033 001102 2$: MOVB #33,$TSTNM ;TEST NUMBER
(3) 042030 012737 042052 001106 MOV #TEST33,$LPADR ;LOAD LOOP ON TEST ADDRESS
(3) 042036 012737 042052 001110 MOV #TEST33,$LPERR ;LOAD LOOP ON ERROR ADDRESS
(1) 042044 012737 007640 001176 MOV #4000,$TIMES ;DO 4000. ITERATIONS
8978 042052 012706 001100 TEST33: MOV #STACK,$P ;LOAD THE STACK POINTER
9015

```

;CLEAR ATTENTION BITS FOR BOTH PORTS

```

(2) 042056 113760 001224 000010 MOVB PORTA,RPCS2(RO) ;SELECT PORT #A
(2) 042064 005060 000012 CLR RPDS1(RO) ;SEIZE THE DRIVE
(2) 042070 012760 000011 000000 MOV #11,RPCS1(RO) ;ISSUE DRIVE CLEAR

```

G10

CZRJECO DL CTRLR LGC MACY11 30A:1052) 28-DEC-77 10:17 PAGE 67-30
CZRJEC.P11 21-DEC-77 14:19 T33

TEST RELEASE THROUGH PORT 'A' WITH ERRORS SET

SEG 0123

```

(042076) 042076 012760 000013 000000      MOV      #13,RPCS1(RO) ;RELEASE THE DRIVE
(042104) 042104 113760 001226 000010      MOV      PORTB,RPCS2(RO) ;SELECT PORT #B
(042112) 042112 005060 000012 000000      CLR      RPDS1(RO) ;SEIZE THE DRIVE THROUGH PORT 'B'
(042116) 042116 012760 000011 000000      MOV      #11,RPCS1(RO) ;ISSUE DRIVE CLEAR
(042124) 042124 012760 000013 000000      MOV      #13,RPCS1(RO) ;RELEASE THE DRIVE
:*****
;SEIZE THE DRIVE THROUGH PORT A
(042132) 042132 113760 001224 000010      MOV      PORTA,RPCS2(RO) ;SELECT PORT A
(042140) 042140 013737 001224 001236      MOV      PORTA,SEIZPT ;STORE SEIZING PORT'S ADDRESS
(042146) 042146 005060 000012 000000      CLR      RPDS1(RO) ;WRITE RPDS1
(042152) 042152 013737 001226 001240      MOV      PORTB,OPPRT ;'OPPOSITE' PORT ADDRESS
:*****
;FORCE AN ERROR
(042160) 042160 012760 177777 000014      MOV      #-1,RPER1(RO) ;SET ERROR BITS
(042166) 042166 012760 000013 000000      MOV      #13,RPCS1(RO) ;ISSUE A RELEASE COMMAND
(042174) 042174 005037 001244 000000      CLR      CKERR ;CLEAR THE 'CHECK ERROR' INDICATOR
(042200) 042200 016037 000000 001126      MOV      RPCS1(RO), $BDDAT ;GET CONTENTS OF RPCS1
(042206) 042206 012737 000000 001122      MOV      #RPCS1,$BDAOR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
(042214) 042214 060037 001122 000000      ADD      RO,$BDAOR ;ADD RH11 BASE ADDRESS
(042220) 042220 012737 004012 001124      MOV      #4012,$GDDAT ;WHAT REGISTER SHOULD BE
(042226) 042226 013737 001126 001164      MOV      $BDDAT,$TMP0 ;MOVE REGISTER CONTENTS TO '$TMP0'
(042234) 042234 042737 173765 001164      BIC      #1C4012,$TMP0 ;SAVE SPECIFIED BITS
(042242) 042242 023737 001124 001164      CMP      $GDDAT,$TMP0 ;COMPARE THE BITS
(042250) 042250 001414 000000 000000      BEQ      66$ ;BR IF OK
(042252) 042252 013737 001126 001174      MOV      $BDDAT,$TMP4 ;COPY 'BAD DATA'
(042260) 042260 042737 004012 001174      BIC      #4012,$TMP4 ;CLEAR THE MASKED BITS
(042266) 042266 053737 001174 001124      BIS      $TMP4,$GDDAT ;'OR' WITH GOOD DATA FOR TYPEOUT
(042274) 042274 104025 000000 000000      ERROR   25 ;TYPE MESSAGE 25
(042276) 042276 005137 001244 000000      COM      CKERR ;SET THE REGISTER COMPARE ERROR INDICATOR
(042302) 042302 000240 000000 000000      NOP
(042304) 042304 005737 001244 000000      TST      CKERR ;DID 'GO' BIT RESET ?
(042310) 042310 001002 000000 000000      BNE      .+6 ;BR IF NOT
(042312) 042312 000137 042352 000000      JMP      1$ ;'GO' BIT RESET
(042316) 042316 012760 000040 000010      MOV      #CLR,RPCS2(RO) ;INIT THE RH11
(042324) 042324 113760 001224 000010      MOV      PORTA,RPCS2(RO) ;SELECT PORT A
(042332) 042332 013737 001224 001234      MOV      PORTA,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
(042340) 042340 012760 000013 000000      MOV      #13,RPCS1(RO) ;RELEASE THE DRIVE THROUGH PORT A
(042346) 042346 000137 043066 000000      JMP      2$ ;BYPASS THE REST OF THE TEST
:*****
;VERIFY THAT DRIVE IS STILL SEIZED BY PORT A
1$:
(042352) 042352 113760 001226 000010      MOV      PORTB,RPCS2(RO) ;SELECT PORT B
(042360) 042360 013737 001226 001234      MOV      PORTB,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
(042366) 042366 005037 001244 000000      CLR      CKERR ;CLEAR THE 'CHECK ERROR' INDICATOR
(042372) 042372 016037 000012 001126      MOV      RPDS1(RO), $BDDAT ;GET CONTENTS OF RPDS1
(042400) 042400 012737 000012 001122      MOV      #RPDS1,$BDAOR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
(042406) 042406 060037 001122 000000      ADD      RO,$BDAOR ;ADD RH11 BASE ADDRESS
(042412) 042412 005037 001124 000000      CLR      $GDDAT ;WHAT REGISTER SHOULD BE
(042416) 042416 023737 001124 001126      CMP      $GDDAT,$BDDAT ;IS THE REGISTER OK ?
(042424) 042424 001403 000000 000000      BEQ      68$ ;BR IF OK

```

H10

02RJEC0, DL CTRLR LGC MACY11 30A,1052
02RJEC.P11 21-DEC-77 14:19

28-DEC-77 10:17 PAGE 67-31
733 TEST RELEASE THROUGH PORT 'A' WITH ERRORS SET

FEQ 0124

```

(042426) 042426 104024 ERRJR 24 ;TYPE MESSAGE 24
(042430) 042430 005137 001244 COM CKERR ;SET THE REGISTER COMPARE ERROR INDICATOR
(042434) 042434 000240 68$: NOP
(042436) 042436 113760 001224 000010 MOVB PORTA,RPCS2(RO) ;SELECT PORT A
(042444) 042444 013737 001224 001234 MOV PORTA,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
(042452) 042452 005037 001244 CLR CKERR ;CLEAR THE 'CHECK ERROR' INDICATOR
(042456) 042456 016037 000014 001126 MOV RPER1(RO), $BDDAT ;GET CONTENTS OF RPER1
(042464) 042464 012737 000014 001122 MOV #RPER1,$B0ADR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
(042472) 042472 060037 001122 ADD RO,$B0ADR ;ADD RH11 BASE ADDRESS
(042476) 042476 012737 177777 001124 MOV #177777,$GDDAT ;WHAT REGISTER SHOULD BE
(042504) 042504 023737 001124 001126 CMP $GDDAT,$BDDAT ;IS THE REGISTER OK ?
(042512) 042512 001403 BEQ 70$ ;BR IF OK
(042514) 042514 104010 ERROR 10 ;REPORT THE ERROR
(042516) 042516 005137 001244 COM CKERR ;SET THE REGISTER COMPARE ERROR INDICATOR
(042522) 042522 000240 70$: NOP

;*****
;CLEAR THE ERRORS THROUGH PORT A

042524 042524 012760 000011 000000 MOV #11,RPCS1(RO) ;ISSUE A DRIVE CLEAR

;*****

;RELEASE THE DRIVE FROM PORT A

042532 042532 113760 001224 000010 MOVB PORTA,RPCS2(RO) ;SELECT PORT A
(042540) 042540 013737 001224 001234 MOV PORTA,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
(042546) 042546 012760 000013 000000 MOV #13,RPCS1(RO) ;ISSUE RELEASE THROUGH PORT A

;VERIFY THAT THE DRIVE IS IN NEUTRAL

042554 042554 005037 001250 CLR RELERR ;CLEAR THE 'RELEASE ERROR' INDICATOR
(042560) 042560 012737 000012 001122 MOV #RPDS1,$B0ADR ;FORM THE ADDRESS OF RPDS1 FOR TYPEOUT
(042566) 042566 060037 001122 ADD RO,$B0ADR ;ADD THE I/O BASE ADDRESS
(042572) 042572 012737 011700 001124 MOV #M0L!PGM!DPR!DRY!VV,$GDDAT ;COMPARISON CONSTANT
(042600) 042600 113760 001224 000010 MOVB PORTA,RPCS2(RO) ;SELECT PORT A.
(042606) 042606 016037 000012 001170 MOV RPDS1(RO), $TMP2 ;GET THE DRIVE STATUS REGISTER FROM PORT A.
(042614) 042614 013737 001170 001164 MOV $TMP2,$TMP0 ;COPY IT INTO '$TMP0'
(042622) 042622 042737 100100 001164 BIC #ATA!VV,$TMP0 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
(042630) 042630 113760 001226 000010 MOVB PORTB,RPCS2(RO) ;SELECT PORT B.
(042636) 042636 016037 000012 001172 MOV RPDS1(RO), $TMP3 ;GET THE DRIVE STATUS REGISTER FROM PORT B.
(042644) 042644 013737 001172 001166 MOV $TMP3,$TMP1 ;COPY IT INTO '$TMP1'
(042652) 042652 042737 100100 001166 BIC #ATA!VV,$TMP1 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
(042660) 042660 023737 001164 001166 CMP $TMP0,$TMP1 ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
(042666) 042666 001006 BNE 72$ ;BR IF NOT
(042670) 042670 005737 001164 TST $TMP0 ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
(042674) 042674 001045 BNE 74$ ;BR IF NOT
(042676) 042676 104046 ERROR 46 ;REPORT DRIVE NOT IN NEUTRAL OR NOT SEIZED
(042700) 042700 000137 043064 JMP 76$ ;BYPASS THE REST OF THE CHECKS
(042704) 042704 013737 001170 001126 72$: MOV $TMP2,$BDDAT ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
(042712) 042712 013737 001226 001234 MOV PORTB,PTNBR ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
(042720) 042720 113760 001226 000010 MOVB PORTB,RPCS2(RO) ;SELECT PORT B.
(042726) 042726 005737 001164 TST $TMP0 ;SEE IF STATUS EQ 0 FROM PORT A.
(042732) 042732 001414 BEQ 73$ ;BR IF ZERO
(042734) 042734 013737 001224 001234 MOV PORTA,PTNBR ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
(042742) 042742 013737 001172 001126 MOV $TMP3,$BDDAT ;'BAD DATA' FOR ERROR TYPE OUT

```

```

(3) 042750 113760 001224 000010      MOV#B  PORTA,RPCS2(RO) ;SELECT PORT A.
(3) 042756 005737 001166              TST    $TMP1           ;SEE IF STATUS EQ ZERO FROM PORT B.
(3) 042762 001012 177777 001250 73$:      BNE    74$            ;BR IF NOT
(3) 042764 012737 177777 001250 73$:      MOV    #-1,RELERR     ;SET 'RELEASE ERROR' INDICATOR
(3) 042772 012760 000011 000000      MOV    #11,RPCS1(RO)  ;CLEAR THE DRIVE
(3) 043000 012760 000013 000000      MOV    #13,RPCS1(RO)  ;RELEASE THE DRIVE
(3) 043006 104026 000013 000000      ERROR  26             ;TYPE ERROR MESSAGE 26
(3) 043010 013737 001170 001126 74$:      MOV    $TMP2,$BDDAT   ;LOOK FOR BIT FAILURES WHEN RPDS1 READ
(3) 043016 013737 001224 001234      MOV    PORTA,PTNBR    ;CHANGE PORT NUMBER
(3) 043024 023737 001124 001170      CMP    $GDDAT,$TMP2   ;ALL BITS OK ?
(3) 043032 001401 000013 000000      BEQ    75$            ;BR IF OK FROM PORT A.
(3) 043034 104007 000013 000000      ERROR  7              ;REPORT ERROR
(3) 043036 013737 001172 001126 75$:      MOV    $TMP3,$BDDAT   ;CHECK RPDS1 FOR BIT FAILURES - FROM PORT B.
(3) 043044 013737 001226 001234      MOV    PORTB,PTNBR    ;CHANGE PORT NUMBER
(3) 043052 023737 001124 001172      CMP    $GDDAT,$TMP3   ;SEE IF READ OK FROM PORT B.
(3) 043060 001401 000013 000000      BEQ    76$            ;BR IF OK
(3) 043062 104007 000013 000000      ERROR  7              ;REPORT ERROR
(3) 043064 000240 000013 000000 76$:      NOP
(3) 043066 000004 000013 000000 2$:      SCOPE                 ;LOOP ?

```

```

*****
*TEST 34      TEST RELEASE THROUGH PORT 'B' WITH ERRORS SET
*
*VERIFY THAT A RELEASE COMMAND PERFORMS NO ACTION IF ISSUED WHEN ERROR
*  BITS ARE SET IN THE DRIVE.
*
*  A. SEIZE THE DRIVE THROUGH PORT 'B' BY WRITING 0'S INTO RPDS1.
*
*  B. WRITE 1'S INTO RPER1 THROUGH PORT 'B'.
*
*  C. ISSUE A RELEASE COMMAND THROUGH PORT 'B'. VERIFY THAT THE 'GO'
*     BIT HAS RESET, THAT THE DRIVE HAS NOT RETURNED TO NEUTRAL, AND
*     THAT RPER1 HAS NOT BEEN CLEARED.
*
*  D. CLEAR RPER1 BY ISSUING A DRIVE CLEAR COMMAND THROUGH PORT 'B'.
*
*  E. ISSUE A RELEASE COMMAND THROUGH PORT 'B'. VERIFY THAT THE DRIVE
*     RETURNED TO NEUTRAL AND THAT NEITHER ATTENTION BIT IS SET.
*
*****

```

```

(3) 043070 005737 001274 000010      TST    KYBCTL          ;PERFORMING ONLY SINGLE TESTS ?
(3) 043074 001406 000013 000000      BEQ    2$              ;BR IF NOT
(3) 043076 100002 000013 000000      BPL    1$              ;BR IF JUST ENTERED TEST
(3) 043100 000137 002622 001274 1$:      JMP    EXEC            ;RETURN & GET NEXT TEST NUMBER
(3) 043104 012737 177777 001274 2$:      MOV    #-1,KYBCTL     ;SET SINGLE TEST INDICATOR
(3) 043112 112737 000034 001102      MOV    #34,$STSTNM    ;TEST NUMBER
(3) 043120 012737 043142 001106      MOV    #TEST34,$LPADR ;LOAD LOOP ON TEST ADDRESS
(3) 043126 012737 043142 001110      MOV    #TEST34,$LPERR ;LOAD LOOP ON ERROR ADDRESS
(1) 043134 012737 007640 001176      MOV    #4000,$TIMES   ;DO 4000. ITERATIONS
9037 043142 012706 001100      TEST34: MOV    #STACK,SP    ;LOAD THE STACK POINTER
9038
;CLEAR ATTENTION BITS FOR BOTH PORTS
(3) 043146 113760 001224 000010      MOV#B  PORTA,RPCS2(RO) ;SELECT PORT #A
(3) 043154 005060 000012 000010      CLR    RPDS1(RO)      ;SEIZE THE DRIVE

```

J10

CZRJECC DL CTRLR LGC MACY11 30A(1052) 28-DEC-77 10:17 PAGE 67-33
CZRJECC.P11 21-DEC-77 14:19 T34

TEST RELEASE THROUGH PORT 'B' WITH ERRORS SET

SEG 0126

```

(2) 043160 012760 000011 000000      MOV      #11,RPCS1(RO)  ;ISSUE DRIVE CLEAR
(2) 043166 012760 000013 000000      MOV      #13,RPCS1(RO)  ;RELEASE THE DRIVE
(2) 043174 113760 001226 000010      MOVVB   PORTB,RPCS2(RO) ;SELECT PORT #B
(2) 043202 005060 000012 000000      CLR      RPDS1(RO)     ;SEIZE THE DRIVE THROUGH PORT 'B'
(2) 043206 012760 000011 000000      MOV      #11,RPCS1(RO)  ;ISSUE DRIVE CLEAR
(2) 043214 012760 000013 000000      MOV      #13,RPCS1(RO)  ;RELEASE THE DRIVE
(2)                                     ;:*****
(2)                                     ;SEIZE THE DRIVE THROUGH PORT B
(2) 043222 113760 001226 000010      MOVVB   PORTB,RPCS2(RO) ;SELECT PORT B
(2) 043230 013737 001226 001236      MOV      PORTB,SEIZPT  ;STORE SEIZING PORT'S ADDRESS
(2) 043236 005060 000012 000000      CLR      RPDS1(RO)     ;WRITE RPDS1
(2) 043242 013737 001224 001240      MOV      PORTA,OPPRT   ;'OPPOSITE' PORT ADDRESS
(2)                                     ;:*****
(2)                                     ;FORCE AN ERROR
(2) 043250 012760 177777 000014      MOV      #-1,RPERR(RO) ;SET ERROR BITS
(2) 043256 012760 000013 000000      MOV      #13,RPCS1(RO) ;ISSUE A RELEASE COMMAND
(2) 043264 005037 001244 000000      CLR      CKERR        ;CLEAR THE 'CHECK ERROR' INDICATOR
(2) 043270 016037 000000 001126      MOV      RPCS1(RO), $BDDAT ;GET CONTENTS OF RPCS1
(2) 043276 012737 000000 001122      MOV      #RPCS1,$BDAOR  ;FORM REGISTER ADDRESS OF ERROR MESSAGE
(2) 043304 060037 001122 000000      ADD     RO,$BDAOR      ;ADD RH11 BASE ADDRESS
(2) 043310 012737 004012 001124      MOV      #4012,$GDDAT  ;WHAT REGISTER SHOULD BE
(2) 043316 013737 001126 001164      MOV      $BDDAT,$TMP0  ;MOVE REGISTER CONTENTS TO '$TMP0'
(2) 043324 042737 173765 001164      BIC     #4012,$TMP0    ;SAVE SPECIFIED BITS
(2) 043332 023737 001124 001164      CMP     $GDDAT,$TMP0   ;COMPARE THE BITS
(2) 043340 001414 000000 000000      BEQ     66$           ;BR IF OK
(2) 043342 013737 001126 001174      MOV     $BDDAT,$TMP4   ;COPY 'BAD DATA'
(2) 043350 042737 004012 001174      BIC     #4012,$TMP4   ;CLEAR THE MASKED BITS
(2) 043356 053737 001174 001124      BIS     $TMP4,$GDDAT  ;'OR' WITH GOOD DATA FOR TYPEOUT
(2) 043364 104025 000000 000000      ERROR  25            ;TYPE MESSAGE 25
(2) 043366 005137 001244 000000      COM     CKERR         ;SET THE REGISTER COMPARE ERROR INDICATOR
(2) 043372 000240 000000 000000      NOP
(2) 043374 005737 001244 000000      66$:  TST     CKERR       ;DID 'GO' BIT RESET ?
(2) 043400 001002 000000 000000      BNE     +6           ;BR IF NOT
(2) 043402 000137 043442 000000      JMP     1$           ;'GO' BIT RESET
(2) 043406 012760 000040 000010      MOV     #CLR,RPCS2(RO) ;INIT THE RH11
(2) 043414 113760 001226 000010      MOVVB   PORTB,RPCS2(RO) ;SELECT PORT B
(2) 043422 013737 001226 001234      MOV     PORTB,PTNBR   ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
(2) 043430 012760 000013 000000      MOV     #13,RPCS1(RO) ;RELEASE THE DRIVE THROUGH PORT B
(2) 043436 000137 044156 000000      JMP     2$           ;BYPASS THE REST OF THE TEST
(2)                                     ;:*****
(2)                                     ;VERIFY THAT DRIVE IS STILL SEIZED BY PORT B
(2) 1$:  MOVVB   PORTA,RPCS2(RO) ;SELECT PORT A
(2) 043442 113760 001224 000010      MOV     PORTA,PTNBR   ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
(2) 043450 013737 001224 001234      CLR     CKERR        ;CLEAR THE 'CHECK ERROR' INDICATOR
(2) 043456 005037 001244 000000      MOV     RPDS1(RO), $BDDAT ;GET CONTENTS OF RPDS1
(2) 043462 016037 000012 001126      MOV     #RPDS1,$BDAOR  ;FORM REGISTER ADDRESS OF ERROR MESSAGE
(2) 043470 012737 000012 001122      ADD     RO,$BDAOR      ;ADD RH11 BASE ADDRESS
(2) 043476 060037 001122 000000      CLR     $GDDAT        ;WHAT REGISTER SHOULD BE
(2) 043502 005037 001124 000000      CMP     $GDDAT,$BDDAT ;IS THE REGISTER OK ?
(2) 043506 023737 001124 001126

```

K10

```

(2) 043514 001403 BEQ 68$ ;BR IF OK
(2) 043516 104024 ERROR 24 ;TYPE MESSAGE 24
(2) 043520 005137 001244 COM CKERR ;SET THE REGISTER COMPARE ERROR INDICATOR
(2) 043524 000240 68$: NOP
(2) 043526 113760 001226 000010 MOVB PORTB,RPCS2(RO) ;SELECT PORT B
(2) 043534 013737 001226 001234 MOV PORTB,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
(2) 043542 005037 001244 CLR CKERR ;CLEAR THE 'CHECK ERROR' INDICATOR
(2) 043546 016037 000014 001126 MOV RPER1(RO), $BDDAT ;GET CONTENTS OF RPER1
(2) 043554 012737 000014 001122 MOV #RPER1,$BDAOR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
(2) 043562 060037 001122 ADD RO,$BDAOR ;ADD RH11 BASE ADDRESS
(2) 043566 012737 177777 001124 MOV #177777,$GDDAT ;WHAT REGISTER SHOULD BE
(2) 043574 023737 001124 001126 CMP $GDDAT,$BDDAT ;IS THE REGISTER OK ?
(2) 043602 001403 BEQ 70$ ;BR IF OK
(2) 043604 104010 ERROR 10 ;REPORT THE ERROR
(2) 043606 005137 001244 COM CKERR ;SET THE REGISTER COMPARE ERROR INDICATOR
(2) 043612 000240 70$: NOP

;*****
;CLEAR THE ERRORS THROUGH PORT B

(1) 043614 012760 000011 000000 MOV #11,RPCS1(RO) ;ISSUE A DRIVE CLEAR

;*****

;RELEASE THE DRIVE FROM PORT B

(3) 043622 113760 001226 000010 MOVB PORTB,RPCS2(RO) ;SELECT PORT B
(3) 043630 013737 001226 001234 MOV PORTB,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
(3) 043636 012760 000013 000000 MOV #13,RPCS1(RO) ;ISSUE RELEASE THROUGH PORT B

;VERIFY THAT THE DRIVE IS IN NEUTRAL

(3) 043644 005037 001250 CLR RELERR ;CLEAR THE 'RELEASE ERROR' INDICATOR
(3) 043650 012737 000012 001122 MOV #RPCS1,$BDAOR ;FORM THE ADDRESS OF RPCS1 FOR TYPEOUT
(3) 043656 060037 001122 ADD RO,$BDAOR ;ADD THE I/O BASE ADDRESS
(3) 043662 012737 011700 001124 MOV #MOL!PGM!DPR!DRY!VV,$GDDAT ;COMPARISON CONSTANT
(3) 043670 113760 001224 000010 MOVB PORTA,RPCS2(RO) ;SELECT PORT A.
(3) 043676 016037 000012 001170 MOV RPS1(RO), $TMP2 ;GET THE DRIVE STATUS REGISTER FROM PORT A.
(3) 043704 013737 001170 001164 MOV $TMP2,$TMP0 ;COPY IT INTO '$TMP0'
(3) 043712 042737 100100 001164 BIC #ATA!VV,$TMP0 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
(3) 043720 113760 001226 000010 MOVB PORTB,RPCS2(RO) ;SELECT PORT B.
(3) 043726 016037 000012 001172 MOV RPS1(RO), $TMP3 ;GET THE DRIVE STATUS REGISTER FROM PORT B.
(3) 043734 013737 001172 001166 MOV $TMP3,$TMP1 ;COPY IT INTO '$TMP1'
(3) 043742 042737 100100 001166 BIC #ATA!VV,$TMP1 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
(3) 043750 023737 001164 001166 CMP $TMP0,$TMP1 ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
(3) 043756 001006 BNE 72$ ;BR IF NOT
(3) 043760 005737 001164 TST $TMP0 ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
(3) 043764 001045 BNE 74$ ;BR IF NOT
(3) 043766 104046 ERROR 46 ;REPORT DRIVE NOT IN NEUTRAL OR NOT SEIZED
(3) 043770 000137 044154 JMP 76$ ;BYPASS THE REST OF THE CHECKS
(3) 043774 013737 001170 001126 72$: MOV $TMP2,$BDDAT ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
(3) 044002 013737 001226 001234 MOV PORTB,PTNBR ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
(3) 044010 113760 001226 000010 MOVB PORTB,RPCS2(RO) ;SELECT PORT B.
(3) 044016 005737 001164 TST $TMP0 ;SEE IF STATUS EQ 0 FROM PORT A.
(3) 044022 001414 BEQ 73$ ;BR IF ZERO
(3) 044024 013737 001224 001234 MOV PORTA,PTNBR ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
    
```


L10

```
(3) 044032 013737 001172 001126 MOV $TMP3,$BDDAT ;'BAD DATA' FOR ERROR TYPE OUT
(3) 044040 113760 001224 000010 MOVB PORTA,APCS2(RO) ;SELECT PORT A.
(3) 044046 005737 001166 TST $TMP1 ;SEE IF STATUS EQ ZERO FROM PORT B.
(3) 044052 001012 BNE 745 ;BR IF NOT
(3) 044054 012737 177777 001250 73$: MOV #-1,RELERR ;SET 'RELEASE ERROR' INDICATOR
(3) 044062 012760 000011 000000 MOV #11,APCS1(RO) ;CLEAR THE DRIVE
(3) 044070 012760 000013 000000 MOV #13,APCS1(RO) ;RELEASE THE DRIVE
(3) 044076 104026 ERROR 26 ;TYPE ERROR MESSAGE 26
(3) 044100 013737 001170 001126 74$: MOV $TMP2,$BDDAT ;LOOK FOR BIT FAILURES WHEN APDS1 READ
(3) 044106 013737 001224 001234 MOV PORTA,PTNBR ;CHANGE PORT NUMBER
(3) 044114 023737 001124 001170 CMP $GDDAT,$TMP2 ;ALL BITS OK ?
(3) 044122 001401 BEQ 755 ;BR IF OK FROM PORT A.
(3) 044124 104007 ERROR 7 ;REPORT ERROR
(3) 044126 013737 001172 001126 75$: MOV $TMP3,$BDDAT ;CHECK APDS1 FOR BIT FAILURES - FROM PORT B.
(3) 044134 013737 001226 001234 MOV PORTB,PTNBR ;CHANGE PORT NUMBER
(3) 044142 023737 001124 001172 CMP $GDDAT,$TMP3 ;SEE IF READ OK FROM PORT B.
(3) 044150 001401 BEQ 765 ;BR IF OK
(3) 044152 104007 ERROR 7 ;REPORT ERROR
(3) 044154 000240 76$: NOP
(1) 044156 000004 2$: SCOPE ;LOOP ?
```

9039
 9055
 9056
 (3)
 (1)
 (1)
 (1)
 (1)
 (1)
 (1)
 (1)
 (1)
 (1)
 (1)
 (1)
 (1)
 (1)
 (1)
 (1)
 (2)
 (3)
 (3)
 (3)
 (3)
 (3)
 (3)
 (3)
 (3)
 (1)
 (1)
 9057
 9097
 (2)
 (2)
 (2)
 (2)
 (2)

```
*****
*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 APDS1.
*
* B. WAIT 500 MS AND WRITE 0'S INTO APDS1 THROUGH PORT 'A'.
*
* C. VERIFY THAT THE TIMEOUT OCCURS WITHIN + OR - 25% OF THE SPECIFIED
* TIME. (THE MEASUREMENT IS MADE FROM STEP 'B'.)
*
* D. VERIFY THAT THE DRIVE RETURNED TO NEUTRAL AND THAT NEITHER ATTENTION
* BIT IS SET.
*****
```

```
↑ST35: TST KYBCTL ;PERFORMING ONLY SINGLE TESTS ?
BEQ 2$ ;BR IF NOT
BPL 1$ ;BR IF JUST ENTERED TEST
JMP EXEC ;RETURN & GET NEXT TEST NUMBER
1$: MOV #-1,KYBCTL ;SET SINGLE TEST INDICATOR
2$: MOVB #35,$STNM ;TEST NUMBER
MOV #TEST35,$LPADR ;LOAD LOOP ON TEST ADDRESS
MOV #TEST35,$LPERR ;LOAD LOOP ON ERROR ADDRESS
MOV #4,$TIMES ;DO 4 ITERATIONS
TEST35: MOV #STACK,SP ;LOAD THE STACK POINTER

;CLEAR ATTENTION BITS FOR BOTH PORTS
MOVB PORTA,APCS2(RO) ;SELECT PORT #A
CLR APDS1(RO) ;SEIZE THE DRIVE
MOV #11,APCS1(RO) ;ISSUE DRIVE CLEAR
MOV #13,APCS1(RO) ;RELEASE THE DRIVE
```

M10

CZRJECO, DL CTRLR LGC MACY11 30A(1052) 28-DEC-77 10:17 PAGE 67-36
CZRJEC.P11 21-DEC-77 14:19

T35 TEST TIMEOUT RETRIGGER THROUGH PORT 'A'

SEQ 0129

```
(2) 044264 113760 001226 000010 MOVB PORTB,RPCS2(RO) ;SELECT PORT #B
(2) 044272 005060 000012 000000 CLR RPOS1(RO) ;SEIZE THE DRIVE THROUGH PORT 'B'
(2) 044276 012760 000011 000000 MOV #11,RPCS1(RO) ;ISSUE DRIVE CLEAR
(2) 044304 012760 000013 000000 MOV #13,RPCS1(RO) ;RELEASE THE DRIVE
(1)
(2)
(2) ;*****
(2) ;SEIZE THE DRIVE THROUGH PORT A
(2) 044312 113760 001224 000010 MOVB PORTA,RPCS2(RO) ;SELECT PORT A
(2) 044320 013737 001224 001236 MOV PORTA,SEIZPT ;STORE SEIZING PORT'S ADDRESS
(2) 044326 005060 000012 000000 CLR RPOS1(RO) ;WRITE RPOS1
(2) 044322 013737 001226 001240 MOV PORTB,OPPRT ;'OPPOSITE' PORT ADDRESS
(1)
(2) ;*****
(1) ;WAIT 500 MS
(1)
(2)
(2) ;*****
(2) ;START THE TIMER
(2) 044340 005037 001252 001254 CLR TIME ;CLEAR THE ELAPSED TIME COUNTER
(2) 044344 012737 000764 001254 MOV #500.,WATCH ;SET WATCH TO 500 MS
(1) 044352 005737 001254 1$: TST WATCH ;WATCH EQUAL TO ZERO
(1) 044356 001375 001254 BNE 1$ ;BR IF NOT
(2)
(2) ;*****
(2) ;START THE TIMER
(2) 044360 005037 001252 001254 CLR TIME ;CLEAR THE ELAPSED TIME COUNTER
(2) 044364 012737 003720 001254 MOV #2000.,WATCH ;SET WATCH TO 2000 MS
(1)
(2) ;*****
(1) ;RETRIGGER THE TIMEOUT ONE-SHOT
(1) 044372 005760 000012 000010 TST RPOS1(RO) ;RETRIGGER THE ONE-SHOT
(2) 044376 113760 001226 000010 MOVB PORTB,RPCS2(RO) ;SELECT PORT B
(2) 044404 013737 001226 001234 MOV PORTB,PINBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
(1) 044412 005760 000012 2$: TST RPOS1(RO) ;WAIT FOR TIMEOUT
(1) 044416 001004 000012 BNE 3$ ;BR IF TIMEOUT OCCURRED
(1) 044420 005737 001254 TST WATCH ;WATCH EQUAL TO ZERO ?
(1) 044424 001372 001254 BNE 2$ ;BR IF NOT
(1) 044426 104036 001252 3$: ERROR 36 ;NO TIMEOUT WITHIN 2 SECONDS
(1) 044430 013737 001252 001272 MOV TIME,TIMES ;SAVE THE ELAPSED TIME VALUE
(1)
(2) ;*****
(2) ;VERIFY THAT THE DRIVE IS IN NEUTRAL
(2) 044436 005037 001250 CLR RELERR ;CLEAR THE 'RELEASE ERROR' INDICATOR
(2) 044442 012737 000012 001122 MOV #RPOS1,$BDADR ;FORM THE ADDRESS OF RPOS1 FOR TYPEOUT
(2) 044450 060037 001122 ADD RO,$BDADR ;ADD THE I/O BASE ADDRESS
(2) 044454 012737 011700 001124 MOV #MOL:PGM:DPR:DRY:VV,$GDDAT ;COMPARISON CONSTANT
(2) 044462 113760 001224 000010 MOVB PORTA,RPCS2(RO) ;SELECT PORT A.
(2) 044470 016037 000012 001170 MOV RPOS1(RO),$TMP2 ;GET THE DRIVE STATUS REGISTER FROM PORT A.
(2) 044476 013737 001170 001164 MOV $TMP2,$TMP0 ;COPY IT INTO '$TMP0'
```

N10

```

(2) 044504 042737 100100 001164 BIC #ATA!VV,$TMP0 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
(2) 044512 113760 001226 000010 MOVB PORTB,$PCS2(RO) ;SELECT PORT B.
(2) 044520 016037 000012 001172 MOV RPS1(RO),$TMP3 ;GET THE DRIVE STATUS REGISTER FROM PORT B.
(2) 044526 013737 001172 001166 MOV $TMP3,$TMP1 ;COPY IT INTO '$TMP1'
(2) 044534 042737 100100 001166 BIC #ATA!VV,$TMP1 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
(2) 044542 023737 001164 001166 CMP $TMP0,$TMP1 ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
(2) 044550 001006 BNE 66$ ;BR IF NOT
(2) 044552 005737 001164 TST $TMP0 ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
(2) 044556 001045 BNE 68$ ;BR IF NOT
(2) 044560 104046 ERROR 46 ;REPORT DRIVE NOT IN NEUTRAL OR NOT SEIZED
(2) 044562 000137 044746 JMP 70$ ;BYPASS THE REST OF THE CHECKS
(2) 044566 013737 001170 001126 66$: MOV $TMP2,$BDDAT ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
(2) 044574 013737 001226 001234 MOV PORTB,$PTNBR ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
(2) 044602 113760 001226 000010 MOVB PORTB,$PCS2(RO) ;SELECT PORT B.
(2) 044610 005737 001164 TST $TMP0 ;SEE IF STATUS EQ 0 FROM PORT A.
(2) 044614 001414 BEQ 67$ ;BR IF ZERO
(2) 044616 013737 001224 001234 MOV PORTA,$PTNBR ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
(2) 044624 013737 001172 001126 MOV $TMP3,$BDDAT ;'BAD DATA' FOR ERROR TYPE OUT
(2) 044632 113760 001224 000010 MOVB PORTA,$PCS2(RO) ;SELECT PORT A.
(2) 044640 005737 001166 TST $TMP1 ;SEE IF STATUS EQ ZERO FROM PORT B.
(2) 044644 001012 BNE 68$ ;BR IF NOT
(2) 044646 012737 177777 001250 67$: MOV #1,$RELEERR ;SET 'RELEASE ERROR' INDICATOR
(2) 044654 012760 000011 000000 MOV #11,$RPS1(RO) ;CLEAR THE DRIVE
(2) 044662 012760 000013 000000 MOV #13,$RPS1(RO) ;RELEASE THE DRIVE
(2) 044670 104022 ERROR 22 ;TYPE ERROR MESSAGE 22
(2) 044672 013737 001170 001126 68$: MOV $TMP2,$BDDAT ;LOOK FOR BIT FAILURES WHEN RPS1 READ
(2) 044700 013737 001224 001234 MOV PORTA,$PTNBR ;CHANGE PORT NUMBER
(2) 044706 023737 001124 001170 CMP $GDDAT,$TMP2 ;ALL BITS OK ?
(2) 044714 001401 BEQ 69$ ;BR IF OK FROM PORT A.
(2) 044716 104007 ERROR 7 ;REPORT ERROR
(2) 044720 013737 001172 001126 69$: MOV $TMP3,$BDDAT ;CHECK RPS1 FOR BIT FAILURES - FROM PORT B.
(2) 044726 013737 001226 001234 MOV PORTB,$PTNBR ;CHANGE PORT NUMBER
(2) 044734 023737 001124 001172 CMP $GDDAT,$TMP3 ;SEE IF READ OK FROM PORT B.
(2) 044742 001401 BEQ 70$ ;BR IF OK
(2) 044744 104007 ERROR 7 ;REPORT ERROR
(2) 044746 000240 70$: NOP
    
```

```

(1) *****
(1) ;CHECK THE TIME FROM RETRIGGER TO TIMEOUT
(1) 044750 023737 001272 001260 CMP TIMES,$TIMEAP ;MEASURED TIME GREATER THAN +25% TOLERANCE ?
(1) 044756 003004 BGT 4$ ;BR IF GREATER
(1) 044760 023737 001272 001262 CMP TIMES,$TIMEAM ;MEASURED TIME LESS THAN -25% TOLERANCE
(1) 044766 002001 BGE .+4 ;BR IF NOT
(1) 044770 104025 4$: ERROR 25 ;REPORT THE ERROR
(1) 044772 000004 SCOPE ;LOOP ?
    
```

```

9113
9114
(3) *****
(4) ;TEST 36 TEST TIMEOUT RETRIGGER THROUGH PORT 'B'
(4) ;
(4) ;VERIFY THAT THE PORT TIMEOUT ONE-SHOT CAN BE RETRIGGERED.
(4) ;
(4) ; A. SEIZE THE DRIVE THROUGH PORT 'B' BY WRITING 0'S INTO RPS1.
(4) ;
(4) ; B. WAIT 500 MS AND WRITE 0'B INTO RPS1 THROUGH PORT 'A'.
(4) ;
    
```

(4) * C. VERIFY THAT THE TIMEOUT OCCURS WITHIN + OR - 25% OF THE SPECIFIED
(4) * TIME. (THE MEASUREMENT IS MADE FROM STEP 'B'.)
(4) *
(4) * D. VERIFY THAT THE DRIVE RETURNED TO NEUTRAL AND THAT NEITHER ATTENTION
(4) * BIT IS SET.
(4) *
(3) *****

```

(2) 044774          ST36:
(3) 044774 005737 001274      TST      KYBCTL      ;PERFORMING ONLY SINGLE TESTS ?
(3) 045000 001406              BEQ      ZS          ;BR IF NOT
(3) 045002 100002              BPL      IS          ;BR IF JUST ENTERED TEST
(3) 045004 000137 002622      JMP      EXEC       ;RETURN & GET NEXT TEST NUMBER
(2) 045010 012737 177777 001274 1S:  MOV     #-1,KYBCTL   ;SET SINGLE TEST INDICATOR
(3) 045016 112737 000036 001102 2S:  MOV     #36,$STNM   ;TEST NUMBER
(3) 045024 012737 045046 001106      MOV     #TEST36,$LPADR ;LOAD LOOP ON TEST ADDRESS
(3) 045032 012737 045046 001110      MOV     #TEST36,$LPERR ;LOAD LOOP ON ERROR ADDRESS
(1) 045040 012737 000004 001176      MOV     #4,$TIMES    ;DO 4 ITERATIONS
9115 045046 012706 001100      TEST36: MOV     #STACK,SP ;LOAD THE STACK POINTER
9116
```

;CLEAR ATTENTION BITS FOR BOTH PORTS

```

(2)
(2) 045052 113760 001224 000010      MOV     PORTA,RPCS2(RO) ;SELECT PORT #A
(2) 045060 005060 000012 000000      CLR     RPDS1(RO)      ;SEIZE THE DRIVE
(2) 045064 012760 000011 000000      MOV     #11,RPCS1(RO)  ;ISSUE DRIVE CLEAR
(2) 045072 012760 000013 000000      MOV     #13,RPCS1(RO)  ;RELEASE THE DRIVE
(2) 045100 113760 001226 000010      MOV     PORTB,RPCS2(RO) ;SELECT PORT #B
(2) 045106 005060 000012 000000      CLR     RPDS1(RO)      ;SEIZE THE DRIVE THROUGH PORT 'B'
(2) 045112 012760 000011 000000      MOV     #11,RPCS1(RO)  ;ISSUE DRIVE CLEAR
(2) 045120 012760 000013 000000      MOV     #13,RPCS1(RO)  ;RELEASE THE DRIVE
```

;*****

;SEIZE THE DRIVE THROUGH PORT B

```

(2) 045126 113760 001226 000010      MOV     PORTB,RPCS2(RO) ;SELECT PORT B
(2) 045134 013737 001226 001236      MOV     PORTB,SEIZPT   ;STORE SEIZING PORT'S ADDRESS
(2) 045142 005060 000012 000000      CLR     RPDS1(RO)      ;WRITE RPDS1
(2) 045146 013737 001224 001240      MOV     PORTA,OPPRT    ;'OPPOSITE' PORT ADDRESS
```

;*****

;WAIT 500 MS

;*****

;START THE TIMER

```

(2) 045154 005037 001252              CLR     TIME          ;CLEAR THE ELAPSED TIME COUNTER
(2) 045160 012737 000764 001254 1S:  MOV     #500.,WATCH   ;SET WATCH TO 500 MS
(1) 045166 005737 001254              TST     WATCH         ;WATCH EQUAL TO ZERO
(1) 045172 001375                      BNE     IS            ;BR IF NOT
```

;*****

;START THE TIMER

```

(2) 045174 005037 001252              CLR     TIME          ;CLEAR THE ELAPSED TIME COUNTER
(2) 045200 012737 003720 001254      MOV     #2000.,WATCH  ;SET WATCH TO 2000 MS
```

```

(1)
(2)
(1)
(1)
(1) 045206 005760 000012          TST      RPDS1(RO)      ;RETRIGGER THE ONE-SHOT
(2) 045212 113760 001224 000010  MOVB    PORTA,RPCS2(RO) ;SELECT PORT A
(2) 045220 013737 001224 001234  MOV     PORTA,PTNBR    ;MOVE PORT ADDRESS TO LOCATION FOR TIMEOUT
(1) 045226 005760 000012          TST      RPDS1(RO)      ;WAIT FOR TIMEOUT
(1) 045232 001004          BNE     3$              ;BR IF TIMEOUT OCCURRED
(1) 045234 005737 001254          TST      WATCH         ;WATCH EQUAL TO ZERO ?
(1) 045240 001372          BNE     2$              ;BR IF NOT
(1) 045242 104036          ERROR   36              ;NO TIMEOUT WITHIN 2 SECONDS
(1) 045244 013737 001252 001272 3$:     MOV     TIME,TIMES     ;SAVE THE ELAPSED TIME VALUE
(1)
(2)
(2)
(2)
(2)
(2) 045252 005037 001250          CLR     RELERR         ;CLEAR THE 'RELEASE ERROR' INDICATOR
(2) 045256 012737 000012 001122  MOV     #RPDS1,$BDDADR ;FORM THE ADDRESS OF RPDS1 FOR TIMEOUT
(2) 045264 060037 001122          ADD     RO,$BDDADR     ;ADD THE I/O BASE ADDRESS
(2) 045270 012737 011700 001124  MOV     #MOL!PGM!DPR!DRY!VV,$GDDAT ;COMPARISON CONSTANT
(2) 045276 113760 001224 000010  MOVB    PORTA,RPCS2(RO) ;SELECT PORT A.
(2) 045304 016037 000012 001170  MOV     RPDS1(RO),STMP2 ;GET THE DRIVE STATUS REGISTER FROM PORT A.
(2) 045312 013737 001170 001164  MOV     STMP2,STMP0     ;COPY IT INTO 'STMP0'
(2) 045320 042737 100100 001164  BIC     #ATA!VV,STMP0   ;CLEAR PORT DEPENDENT BITS FROM THE COPY
(2) 045326 113760 001226 000010  MOVB    PORTB,RPCS2(RO) ;SELECT PORT B.
(2) 045334 016037 000012 001172  MOV     RPDS1(RO),STMP3 ;GET THE DRIVE STATUS REGISTER FROM PORT B.
(2) 045342 013737 001172 001166  MOV     STMP3,STMP1     ;COPY IT INTO 'STMP1'
(2) 045350 042737 100100 001166  BIC     #ATA!VV,STMP1   ;CLEAR PORT DEPENDENT BITS FROM THE COPY
(2) 045356 023737 001164 001166  CMP     STMP0,STMP1     ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
(2) 045364 001006          BNE     66$            ;BR IF NOT
(2) 045366 005737 001164          TST     STMP0          ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
(2) 045372 001045          BNE     68$            ;BR IF NOT
(2) 045374 104046          ERROR   46              ;REPORT DRIVE NOT IN NEUTRAL OR NOT SEIZED
(2) 045376 000137 045562          JMP     70$            ;BYPASS THE REST OF THE CHECKS
(2) 045402 013737 001170 001126 66$:   MOV     STMP2,$BDDAT   ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
(2) 045410 013737 001226 001234  MOV     PORTB,PTNBR    ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
(2) 045416 113760 001226 000010  MOVB    PORTB,RPCS2(RO) ;SELECT PORT B.
(2) 045424 005737 001164          TST     STMP0          ;SEE IF STATUS EQ 0 FROM PORT A.
(2) 045430 001414          BEQ     67$            ;BR IF ZERO
(2) 045432 013737 001224 001234  MOV     PORTA,PTNBR    ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
(2) 045440 013737 001172 001126  MOV     STMP3,$BDDAT   ;'BAD DATA' FOR ERROR TYPE OUT
(2) 045446 113760 001224 000010  MOVB    PORTA,RPCS2(RO) ;SELECT PORT A.
(2) 045454 005737 001166          TST     STMP1          ;SEE IF STATUS EQ ZERO FROM PORT B.
(2) 045460 001012          BNE     68$            ;BR IF NOT
(2) 045462 012737 177777 001250 67$:   MOV     #-1,RELERR     ;SET 'RELEASE ERROR' INDICATOR
(2) 045470 012760 000011 000000  MOV     #11,RPCS1(RO)  ;CLEAR THE DRIVE
(2) 045476 012760 000013 000000  MOV     #13,RPCS1(RO)  ;RELEASE THE DRIVE
(2) 045504 104022          ERROR   22              ;TYPE ERROR MESSAGE 22
(2) 045506 013737 001170 001126 68$:   MOV     STMP2,$BDDAT   ;LOOK FOR BIT FAILURES WHEN RPDS1 READ
(2) 045514 013737 001224 001234  MOV     PORTA,PTNBR    ;CHANGE PORT NUMBER
(2) 045522 023737 001124 001170  CMP     $GDDAT,STMP2   ;ALL BITS OK ?
(2) 045530 001401          BEQ     69$            ;BR IF OK FROM PORT A.
(2) 045532 104007          ERROR   7              ;REPORT ERROR
(2) 045534 013737 001172 001126 69$:   MOV     STMP3,$BDDAT   ;CHECK RPDS1 FOR BIT FAILURES - FROM PORT B.

```

```

(2) 045542 013737 001226 001234      MOV      PORTB,PTNBR      ;CHANGE PORT NUMBER
(2) 045550 023737 001124 001172      CMP      %GDAT,%STMP3    ;SEE IF READ OK FROM PORT B.
(2) 045556 001401                       BEQ      70$             ;BR IF OK
(2) 045560 104007                       ERROR    7              ;REPORT ERROR
(2) 045562 000240                       70$:    NOP
(1)
(2)
(1)
(1) 045564 023737 001272 001266      CMP      TIMES,TIMEBP    ;MEASURED TIME GREATER THAN +25% TOLERANCE ?
(1) 045572 003004                       BGT     4$              ;BR IF GREATER
(1) 045574 023737 001272 001270      CMP      TIMES,TIMEBM    ;MEASURED TIME LESS THAN -25% TOLERANCE
(1) 045602 002001                       BGE     +4             ;BR IF NOT
(1) 045604 104025                       4$:    ERROR    25      ;REPORT THE ERROR
(1) 045606 000004                       SCOPE   ;LOOP ?

```

9117
9118
9135
9136
(3)
(4)
(4)
(4)
(4)
(4)
(4)
(4)
(4)
(4)
(4)
(4)
(2)
(3)
(3)
(3)
(3)
(3)
(3)
(3)
(3)
(3)
(1)
9137
9169
(2)
(2)
(2)
(2)
(2)
(2)
(2)
(2)
(2)

```

*****
*TEST 37      TEST PORT 'A' ATTENTION AFTER A COMMAND
*
*TEST THE OPERATION OF THE PORT A AND PORT B ATTENTION BITS AFTER A
*COMMAND.
*
*  A.  ISSUE A RECALIBRATE COMMAND THROUGH PORT 'A'.
*
*  B.  WAIT FOR THE RECALIBRATE COMMAND TO COMPLETE ('DRY' TO BECOME
*       '1').  VERIFY THAT THE ATTENTION BIT FOR PORT 'A' IS SET AND
*       THAT THE ATTENTION BIT FOR PORT 'B' IS NOT SET.
*
*  C.  RELEASE THE DRIVE THROUGH PORT 'A'.  VERIFY THAT THE DRIVE RETURNED
*       TO NEUTRAL AND THAT NEITHER ATTENTION BIT IS SET.
*****

```

```

†ST37:
      TST      KYBCTL      ;PERFORMING ONLY SINGLE TESTS ^
      BEQ      2$         ;BR IF NOT
      BPL     1$         ;BR IF JUST ENTERED TEST
      JMP     EXEC        ;RETURN & GET NEXT TEST NUMBER
1$:   MOV     #-1,KYBCTL   ;SET SINGLE TEST INDICATOR
2$:   MOVB   #37,%STNM    ;TEST NUMBER
      MOV     #TEST37,%SLPADR ;LOAD LOOP ON TEST ADDRESS
      MOV     #TEST37,%SLPERR ;LOAD LOOP ON ERROR ADDRESS
      MOV     #4,%TIMES   ;DO 4 ITERATIONS
TEST37: MOV    #STACK,%SP ;LOAD THE STACK POINTER

```

```

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

```

E11

```

(2) 045734 012760 000013 000000      MOV      #13,RPCS1(RO) ;RELEASE THE DRIVE
(2) 045742 113760 001224 000010      MOVB     PORTA,RPCS2(RO) ;SELECT PORT A
(2) 045750 013737 001224 001234      MOV      PORTA,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
(1) 045756 013737 001224 001236      MOV      PORTA,SEIZPT ;'SEIZED' PORT ADDRESS
(1)
(2)
(1)
(1) 045764 012760 000007 000000      MOV      #7,RPCS1(RO) ;ISSUE A RECALIBRATE INSTRUCTION THROUGH PORT A
(1)
(2)
(1)
(1) 045772 032760 000200 000012      BIT      #DRY,RPDS1(RO) ;WAIT FOR DRIVE TO FINISH
(1) 046000 001774                      BEQ      .-6 ;BR IF NOT FINISHED
(1)
(2)
(1)
(1)
(2) 046002 005037 001244                      CLR      CKERR ;CLEAR THE 'CHECK ERROR' INDICATOR
(2) 046006 016037 000012 001126      MOV      RPDS1(RO), $BDDAT ;GET CONTENTS OF RPDS1
(2) 046014 012737 000012 001122      MOV      #RPDS1,$B0ADR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
(2) 046022 060037 001122                      ADD      RO,$B0ADR ;ADD RH11 BASE ADDRESS
(2) 046026 012737 100000 001124      MOV      #ATA,$GDDAT ;WHAT REGISTER SHOULD BE
(2) 046034 013737 001126 001164      MOV      $BDDAT,$TMP0 ;MOVE REGISTER CONTENTS TO '$TMP0'
(2) 046042 042737 077777 001164      BIC      #ICATA,$TMP0 ;SAVE SPECIFIED BITS
(2) 046050 023737 001124 001164      CMP      $GDDAT,$TMP0 ;COMPARE THE BITS
(2) 046056 001414                      BEQ      64$ ;BR IF OK
(2) 046060 013737 001126 001174      MOV      $BDDAT,$TMP4 ;COPY 'BAD DATA'
(2) 046066 042737 100000 001174      BIC      #ATA,$TMP4 ;CLEAR THE MASKED BITS
(2) 046074 053737 001174 001124      BIS      $TMP4,$GDDAT ;'OR' WITH GOOD DATA FOR TYPEOUT
(2) 046102 104032                      ERROR    32 ;TYPE MESSAGE 32
(2) 046104 005137 001244                      COM      CKERR ;SET THE REGISTER COMPARE ERROR INDICATOR
(2) 046110 000240                      NOP
(2) 046112 113760 001226 000010      MOVB     PORTB,RPCS2(RO) ;SELECT PORT B
(2) 046120 013737 001226 001234      MOV      PORTB,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
(1)
(2)
(1)
(1)
(2)
(1)
(1)
(2) 046126 005037 001244                      CLR      CKERR ;CLEAR THE 'CHECK ERROR' INDICATOR
(2) 046132 016037 000012 001126      MOV      RPDS1(RO), $BDDAT ;GET CONTENTS OF RPDS1
(2) 046140 012737 000012 001122      MOV      #RPDS1,$B0ADR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
(2) 046146 060037 001122                      ADD      RO,$B0ADR ;ADD RH11 BASE ADDRESS
(2) 046152 005037 001124                      CLR      $GDDAT ;WHAT REGISTER SHOULD BE
(2) 046156 013737 001126 001164      MOV      $BDDAT,$TMP0 ;MOVE REGISTER CONTENTS TO '$TMP0'
(2) 046164 042737 077777 001164      BIC      #ICATA,$TMP0 ;SAVE SPECIFIED BITS
(2) 046172 023737 001124 001164      CMP      $GDDAT,$TMP0 ;COMPARE THE BITS
(2) 046200 001414                      BEQ      66$ ;BR IF OK
(2) 046202 013737 001126 001174      MOV      $BDDAT,$TMP4 ;COPY 'BAD DATA'
(2) 046210 042737 100000 001174      BIC      #ATA,$TMP4 ;CLEAR THE MASKED BITS
(2) 046216 053737 001174 001124      BIS      $TMP4,$GDDAT ;'OR' WITH GOOD DATA FOR TYPEOUT
(2) 046224 104032                      ERROR    32 ;TYPE MESSAGE 32
(2) 046226 005137 001244                      COM      CKERR ;SET THE REGISTER COMPARE ERROR INDICATOR
(2) 046232 000240                      NOP
6c3:

```

F11

::*****

```

(2) ;RELEASE THE DRIVE FROM PORT A
(2)
(2)
(2) 046234 113760 001224 000010 MOV#B PORTA,RPCS2(RO) ;SELECT PORT A
(3) 046242 013737 001224 001234 MOV PORTA,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
(2) 046250 012760 000013 000000 MOV #13,RPCS1(RO) ;ISSUE RELEASE THROUGH PORT A
(3)
(3) ;VERIFY THAT THE DRIVE IS IN NEUTRAL
(3)
(3) 046256 005037 001250 CLR RELERR ;CLEAR THE 'RELEASE ERROR' INDICATOR
(3) 046262 012737 000012 001122 MOV #RPOS1,$BDDADR ;FORM THE ADDRESS OF RPOS1 FOR TYPEOUT
(3) 046270 060037 001122 ADD RO,$BDDADR ;ADD THE I/O BASE ADDRESS
(3) 046274 012737 011700 001124 MOV #MOL!V!M!DPR!DRY!V!V,$GDDAT ;COMPARISON CONSTANT
(3) 046302 113760 001224 000010 MOV#B PORTA,RPCS2(RO) ;SELECT PORT A.
(3) 046310 016037 000012 001170 MOV RPOS1(RO),$TMP2 ;GET THE DRIVE STATUS REGISTER FROM PORT A.
(3) 046316 013737 001170 001164 MOV $TMP2,$TMP0 ;COPY IT INTO '$TMP0'
(3) 046324 042737 100100 001164 BIC #ATA!V!V,$TMP0 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
(3) 046332 113760 001226 000010 MOV#B PORTB,RPCS2(RO) ;SELECT PORT B.
(3) 046340 016037 000012 001172 MOV RPOS1(RO),$TMP3 ;GET THE DRIVE STATUS REGISTER FROM PORT B.
(3) 046346 013737 001172 001166 MOV $TMP3,$TMP1 ;COPY IT INTO '$TMP1'
(3) 046354 042737 100100 001166 BIC #ATA!V!V,$TMP1 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
(3) 046362 023737 001164 001166 CMP $TMP0,$TMP1 ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS
(3) 046370 001006 BNE 68$ ;BR IF NOT
(3) 046372 005737 001164 TST $TMP0 ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
(3) 046376 001045 BNE 70$ ;BR IF NOT
(3) 046400 104046 ERROR 46 ;REPORT DRIVE NOT IN NEUTRAL OR NOT SEIZED
(3) 046402 000137 046566 JMP 72$ ;BYPASS THE REST OF THE CHECKS
(3) 046406 013737 001170 001126 68$: MOV $TMP2,$BDDAT ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
(3) 046414 013737 001226 001234 MOV PORTB,PTNBR ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
(3) 046422 113760 001226 000010 MOV#B PORTB,RPCS2(RO) ;SELECT PORT B.
(3) 046430 005737 001164 TST $TMP0 ;SEE IF STATUS EQ 0 FROM PORT A.
(3) 046434 001414 BEQ 69$ ;BR IF ZERO
(3) 046436 013737 001224 001234 MOV PORTA,PTNBR ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
(3) 046444 013737 001172 001126 MOV $TMP3,$BDDAT ;'BAD DATA' FOR ERROR TYPE OUT
(3) 046452 113760 001224 000010 MOV#B PORTA,RPCS2(RO) ;SELECT PORT A.
(3) 046460 005737 001166 TST $TMP1 ;SEE IF STATUS EQ ZERO FROM PORT B.
(3) 046464 001012 BNE 70$ ;BR IF NOT
(3) 046466 012737 177777 001250 69$: MOV #-1,RELERR ;SET 'RELEASE ERROR' INDICATOR
(3) 046474 012760 000011 000000 MOV #11,RPCS1(RO) ;CLEAR THE DRIVE
(3) 046502 012760 000013 000000 MOV #13,RPCS1(RO) ;RELEASE THE DRIVE
(3) 046510 104026 ERROR 26 ;TYPE ERROR MESSAGE 26
(3) 046512 013737 001170 001126 70$: MOV $TMP2,$BDDAT ;LOOK FOR BIT FAILURES WHEN RPOS1 READ
(3) 046520 013737 001224 001234 MOV PORTA,PTNBR ;CHANGE PORT NUMBER
(3) 046526 023737 001124 001170 CMP $GDDAT,$TMP2 ;ALL BITS OK ?
(3) 046534 001401 BEQ 71$ ;BR IF OK FROM PORT A.
(3) 046536 104007 ERROR 7 ;REPORT ERROR
(3) 046540 013737 001172 001126 71$: MOV $TMP3,$BDDAT ;CHECK RPOS1 FOR BIT FAILURES - FROM PORT B.
(3) 046546 013737 001226 001234 MOV PORTB,PTNBR ;CHANGE PORT NUMBER
(3) 046554 023737 001124 001172 CMP $GDDAT,$TMP3 ;SEE IF READ OK FROM PORT B.
(3) 046562 001401 BEQ 72$ ;BR IF OK
(3) 046564 104007 ERROR 7 ;REPORT ERROR
(3) 046566 000240 72$: NOP
(3) 046570 000004 SCOPE ;LOOP ?

```

::*****

3:05
3:06


```

*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.

```

```

S40: TST KYBCTL ;PERFORMING ONLY SINGLE TESTS
      BEQ 2$ ;BR IF NOT
      BPL 1$ ;BR IF JUST ENTERED TEST
      JMP EXEC ;RETURN & GET NEXT TEST NUMBER
1$: MOV #-1,KYBCTL ;SET SINGLE TEST INDICATOR
2$: MOVB #40,$STSTM ;TEST NUMBER
      MOV #TEST40,$LPADR ;LOAD LOOP ON TEST ADDRESS
      MOV #TEST40,$LPERR ;LOAD LOOP ON ERROR ADDRESS
      MOV #4,$TIMES ;DO 4 ITERATIONS
TEST40: MOV #STACK,$SP ;LOAD THE STACK POINTER

```

;CLEAR ATTENTION BITS FOR BOTH PORTS

```

MOVB PORTA,$RPCS2(R0) ;SELECT PORT #A
CLR $RPS1(R0) ;SEIZE THE DRIVE
MOV #11,$RPCS1(R0) ;ISSUE DRIVE CLEAR
MOV #13,$RPCS1(R0) ;RELEASE THE DRIVE
MOVB PORTB,$RPCS2(R0) ;SELECT PORT #B
CLR $RPS1(R0) ;SEIZE THE DRIVE THROUGH PORT 'B'
MOV #11,$RPCS1(R0) ;ISSUE DRIVE CLEAR
MOV #13,$RPCS1(R0) ;RELEASE THE DRIVE
MOVB PORTB,$RPCS2(R0) ;SELECT PORT B
MOV PORTB,$PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR 'TYPEOUT'
MOV PORTB,$SEIZPT ;'SEIZED' PORT ADDRESS

```

;DO A RECALIBRATE THROUGH PORT B
MOV #7,\$RPCS1(R0) ;ISSUE A RECALIBRATE INSTRUCTION THROUGH PORT B

;WAIT FOR DRIVE TO FINISH

```

BIT #DRY,$RPS1(R0) ;WAIT FOR DRIVE TO FINISH
BEQ -6 ;BR IF NOT FINISHED

```

;CONFIRM THAT ATTENTION IS SET FOR PORT B

CLR CKERP ;CLEAR THE 'CHECK ERROR' INDICATOR

046572
046576
046600
046602
046606
046614
046622
046630
046636
046644

046650
046656
046662
046670
046676
046704
046710
046716
046724
046732
046740

046746

046754
046762

046764

```

046572 005737 001274
046576 001406
046600 100002
046602 000137 002622
046606 012737 177777 001274
046614 112737 000040 001102
046622 012737 046644 001106
046630 012737 046644 001110
046636 012737 000004 001176
046644 012706 001100

046650 113760 001224 000010
046656 005060 000012
046662 012760 000011 000000
046670 012760 000013 000000
046676 113760 001226 000010
046704 005060 000012
046710 012760 000011 000000
046716 012760 000013 000000
046724 113760 001226 000010
046732 013737 001226 001234
046740 013737 001226 001236

046746 012760 000007 000000

046754 032760 000200 000012
046762 001774

046764 005037 001244

```

046770	016037	000012	001126	MOV	RPDS1(RO), \$BDDAT ; GET CONTENTS OF RPDS1
046776	012737	000012	001122	MOV	#RPDS1, \$BDAOR ; FORM REGISTER ADDRESS OF ERROR MESSAGE
047004	060037	001122		ADD	RO, \$BDAOR ; ADD RH11 BASE ADDRESS
047010	012737	100000	001124	MOV	#ATA, \$GDDAT ; WHAT REGISTER SHOULD BE
047016	013737	001126	001164	MOV	\$BDDAT, \$TMP0 ; MOVE REGISTER CONTENTS TO '\$TMP0'
047024	042737	077777	001164	BIC	#1CATA, \$TMP0 ; SAVE SPECIFIED BITS
047032	023737	001124	001164	CMP	\$GDDAT, \$TMP0 ; COMPARE THE BITS
047040	001414			BEQ	64\$; BR IF OK
047042	013737	001126	001174	MOV	\$BDDAT, \$TMP4 ; COPY 'BAD DATA'
047050	042737	100000	001174	BIC	#ATA, \$TMP4 ; CLEAR THE MASKED BITS
047056	053737	001174	001124	BIS	\$TMP4, \$GDDAT ; 'OR' WITH GOOD DATA FOR TYPEOUT
047064	104032			ERROR	32 ; TYPE MESSAGE 32
047066	005137	001244		COM	CKERR ; SET THE REGISTER COMPARE ERROR INDICATOR
047072	000240			64\$:	NOP
047074	113760	001224	000010	MOVB	PORTA, RPCS2(RO) ; SELECT PORT A
047102	013737	001224	001234	MOV	PORTA, PTNBR ; MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT

;CONFIRM THAT ATTENTION IS NOT SET FOR PORT A

047110	005037	001244		CLR	CKERR ; CLEAR THE 'CHECK ERROR' INDICATOR
047114	016037	000012	001126	MOV	RPDS1(RO), \$BDDAT ; GET CONTENTS OF RPDS1
047122	012737	000012	001122	MOV	#RPDS1, \$BDAOR ; FORM REGISTER ADDRESS OF ERROR MESSAGE
047130	060037	001122		ADD	RO, \$BDAOR ; ADD RH11 BASE ADDRESS
047134	005037	001124		CLR	\$GDDAT ; WHAT REGISTER SHOULD BE
047140	013737	001126	001164	MOV	\$BDDAT, \$TMP0 ; MOVE REGISTER CONTENTS TO '\$TMP0'
047146	042737	077777	001164	BIC	#1CATA, \$TMP0 ; SAVE SPECIFIED BITS
047154	023737	001124	001164	CMP	\$GDDAT, \$TMP0 ; COMPARE THE BITS
047162	001414			BEQ	66\$; BR IF OK
047164	013737	001126	001174	MOV	\$BDDAT, \$TMP4 ; COPY 'BAD DATA'
047172	042737	100000	001174	BIC	#ATA, \$TMP4 ; CLEAR THE MASKED BITS
047200	053737	001174	001124	BIS	\$TMP4, \$GDDAT ; 'OR' WITH GOOD DATA FOR TYPEOUT
047206	104032			ERROR	32 ; TYPE MESSAGE 32
047210	005137	001244		COM	CKERR ; SET THE REGISTER COMPARE ERROR INDICATOR
047214	000240			66\$:	NOP

;RELEASE THE DRIVE FROM PORT B

047216	113760	001226	000010	MOVB	PORTB, RPCS2(RO) ; SELECT PORT B
047224	013737	001226	001234	MOV	PORTB, PTNBR ; MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
047232	012760	000013	000000	MOV	#13, RPCS1(RO) ; ISSUE RELEASE THROUGH PORT B

;VERIFY THAT THE DRIVE IS IN NEUTRAL

047240	005037	001250		CLR	RELERR ; CLEAR THE 'RELEASE ERROR' INDICATOR
047244	012737	000012	001122	MOV	#RPDS1, \$BDAOR ; FORM THE ADDRESS OF RPDS1 FOR TYPEOUT
047252	060037	001122		ADD	RO, \$BDAOR ; ADD THE I/O BASE ADDRESS
047256	012737	011700	001124	MOV	#MOL:PGM:DPR:DRY:VV, \$GDDAT ; COMPARISON CONSTANT
047264	113760	001224	000010	MOVB	PORTA, RPCS2(RO) ; SELECT PORT A
047272	016037	000012	001170	MOV	RPDS1(RO), \$TMP2 ; GET THE DRIVE STATUS REGISTER FROM PORT A.
047300	013737	001170	001164	MOV	\$TMP2, \$TMP0 ; COPY IT INTO '\$TMP0'
047306	042737	100100	001164	BIC	#ATA!VV, \$TMP0 ; CLEAR PORT DEPENDENT BITS FROM THE COPY
047314	113760	001226	000010	MOVB	PORTB, RPCS2(RO) ; SELECT PORT B.
047322	016037	000012	001172	MOV	RPDS1(RO), \$TMP3 ; GET THE DRIVE STATUS REGISTER FROM PORT B.

```

(3) 047330 013737 001172 001166      MOV      $TMP3,$TMP1      ;COPY IT INTO '$TMP1'
(3) 047336 042737 100100 001166      BIC      @ATA!VV,$TMP1    ;CLEAR PORT DEPENDENT BITS FROM THE COPY
(3) 047344 023737 001164 001166      CMP      $TMP0,$TMP1     ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
(3) 047352 001006      BNE      68$             ;BR IF NOT
(3) 047354 005737 001164      TST      $TMP0           ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
(3) 047360 001045      BNE      70$             ;BR IF NOT
(3) 047362 104046      ERROR   46             ;REPORT DRIVE NOT IN NEUTRAL OR NOT SEIZED
(3) 047364 000137 047550      JMP      72$             ;BYPASS THE REST OF THE CHECKS
(3) 047370 013737 001170 001126 68$:  MOV      $TMP2,$BDDAT    ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
(3) 047376 013737 001226 001234      MOV      PORTB,PTNBR     ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
(3) 047404 113760 001226 000010      MOV      PORTB,RPCS2(RO) ;SELECT PORT B.
(3) 047412 005737 001164      TST      $TMP0           ;SEE IF STATUS EQ 0 FROM PORT A.
(3) 047416 001414      BEQ      69$             ;BR IF ZERO
(3) 047420 013737 001224 001234      MOV      PORTA,PTNBR     ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
(3) 047426 013737 001172 001126      MOV      $TMP3,$BDDAT    ;'BAD DATA' FOR ERROR TYPE OUT
(3) 047434 113760 001224 000010      MOV      PORTA,RPCS2(RO) ;SELECT PORT A.
(3) 047442 005737 001166      TST      $TMP1           ;SEE IF STATUS EQ ZERO FROM PORT B.
(3) 047446 001012      BNE      70$             ;BR IF NOT
(3) 047450 012737 177777 001250 69$:  MOV      #-1,RELERR      ;SET 'RELEASE ERROR' INDICATOR
(3) 047456 012760 000011 000000      MOV      #11,RPCS1(RO)   ;CLEAR THE DRIVE
(3) 047464 012760 000013 000000      MOV      #13,RPCS1(RO)   ;RELEASE THE DRIVE
(3) 047472 104026      ERROR   26             ;TYPE ERROR MESSAGE 26
(3) 047474 013737 001170 001126 70$:  MOV      $TMP2,$BDDAT    ;LOOK FOR BIT FAILURES WHEN RPDS1 PEAC
(3) 047502 013737 001224 001234      MOV      PORTA,PTNBR     ;CHANGE PORT NUMBER
(3) 047510 023737 001124 001170      CMP      $GDDAT,$TMP2    ;ALL BITS OK ?
(3) 047516 001401      BEQ      71$             ;BR IF OK FROM PORT A.
(3) 047520 104007      ERROR   7              ;REPORT ERROR
(3) 047522 013737 001172 001126 71$:  MOV      $TMP3,$BDDAT    ;CHECK RPDS1 FOR BIT FAILURES - FROM PORT B.
(3) 047530 013737 001226 001234      MOV      PORTB,PTNBR     ;CHANGE PORT NUMBER
(3) 047536 023737 001124 001172      CMP      $GDDAT,$TMP3    ;SEE IF READ OK FROM PORT B.
(3) 047544 001401      BEQ      72$             ;BR IF OK
(3) 047546 104007      ERROR   7              ;REPORT ERROR
(3) 047550 000240      NOP
(3) 047552 000004      SCOPE

```

(1) 047552
(1) 047546
(1) 047544
(1) 047536
(1) 047530
(1) 047522
(1) 047520
(1) 047516
(1) 047510
(1) 047502
(1) 047474
(1) 047472
(1) 047464
(1) 047456
(1) 047450
(1) 047446
(1) 047442
(1) 047434
(1) 047426
(1) 047420
(1) 047416
(1) 047412
(1) 047404
(1) 047376
(1) 047370
(1) 047364
(1) 047362
(1) 047354
(1) 047352
(1) 047344
(1) 047336
(1) 047330

```

*****
*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.
*
*****

```

```

(4)
(4)
(4)
(4)
(3)
(2) 047554
(3) 047554 005737 001274
(3) 047560 001406
(3) 047562 100002
(3) 047564 000137 002622
(3) 047570 012737 177777 001274 1$:
(3) 047576 112737 000041 001102 2$:
(3) 047604 012737 047626 001106
(3) 047612 012737 047626 001110
(1) 047620 012737 007640 001176
9216 047626 012706 001100 TEST41:
9251
(2)
(2)
(2) 047632 113760 001224 000010
(2) 047640 005060 000012
(2) 047644 012760 000011 000000
(2) 047652 012760 000013 000000
(2) 047660 113760 001226 000010
(2) 047666 005060 000012
(2) 047672 012760 000011 000000
(2) 047700 012760 000013 000000
(2)
(2)
(2)
(2) 047706 113760 001224 000010
(2) 047714 013737 001224 001236
(2) 047722 005060 000012
(2) 047726 013737 001226 001240
(2) 047734 012760 177777 000014
(2) 047742 012760 177777 000040
(2) 047750 012760 177777 000042
(2) 047756 113760 001226 000010
(2) 047764 013737 001226 001234
(1) 047772 004737 050630
(2) 047776 113760 001224 000010
(2) 050004 013737 001224 001234
(2) 050012 005060 000042
(2) 050016 005060 000040
(2) 050022 005060 000014
(1) 050026 013760 001232 000016
(2) 050034 113760 001226 000010
(2) 050042 013737 001226 001234
(2) 050050 012760 177777 000014
(2) 050056 012760 177777 000040
(2) 050064 012760 177777 000042
(2) 050072 113760 001224 000010
(2) 050100 013737 001224 001234
(1) 050106 004737 050630
(2)
(2)

```

```

:
: * G. ISSUE A RELEASE COMMAND THROUGH PORT 'B'. VERIFY THAT THE DRIVE
: * RETURNED TO NEUTRAL AND THAT NEITHER ATTENTION BIT IS SET.
:
: *****
TST41:
TST KYBCTL ;PERFORMING ONLY SINGLE TESTS ?
BEQ 2$ ;BR IF NOT
BPL 1$ ;BR IF JUST ENTERED TEST
JMP EXEC ;RETURN & GET NEXT TEST NUMBER
1$: MOV #-1,KYBCTL ;SET SINGLE TEST INDICATOR
2$: MOVB #41,$STNM ;TEST NUMBER
MOV #TEST41,$LPADR ;LOAD LOOP ON TEST ADDRESS
MOV #TEST41,$LPERR ;LOAD LOOP ON ERROR ADDRESS
MOV #4000,$TIMES ;DO 4000 ITERATIONS
TEST41: MOV #STACK,$SP ;LOAD THE STACK POINTER

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

;SEIZE THE DRIVE THROUGH PORT A
MOVB PORTA,RPCS2(RO) ;SELECT PORT A
MOV PORTA,SEIZPT ;STORE SEIZING PORT'S ADDRESS
CLR RPDS1(RO) ;WRITE RPDS1
MOV PORTB,OPPRT ;'OPPOSITE' PORT ADDRESS
MOV #-1,RPER1(RO) ;LOAD 1'S INTO RPER1 THROUGH PORT A
MOV #-1,RPER2(RO) ;LOAD 1'S INTO RPER2 THROUGH PORT A
MOV #-1,RPER3(RO) ;LOAD 1'S INTO RPER3 THROUGH PORT A
MOVB PORTB,RPCS2(RO) ;SELECT PORT B
MOV PORTB,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
JSR PC,TST41B ;CHECK THE REGISTERS THROUGH PORT B
MOVB PORTA,RPCS2(RO) ;SELECT PORT A
MOV PORTA,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
CLR RPER3(RO) ;CLEAR RPER3 ON PORT A
CLR RPER2(RO) ;CLEAR RPER2 ON PORT A
CLR RPER1(RO) ;CLEAR RPER1 ON PORT A
MOV ASR1,RPAS(RO) ;CLEAR THE ATTENTION BIT FOR PORT A
MOVB PORTB,RPCS2(RO) ;SELECT PORT B
MOV PORTB,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
MOV #-1,RPER1(RO) ;LOAD 1'S INTO RPER1 THROUGH PORT B
MOV #-1,RPER2(RO) ;LOAD 1'S INTO RPER2 THROUGH PORT B
MOV #-1,RPER3(RO) ;LOAD 1'S INTO RPER3 THROUGH PORT B
MOVB PORTA,RPCS2(RO) ;SELECT PORT A
MOV PORTA,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
JSR PC,TST41B ;CHECK THE REGISTERS THROUGH PORT A

;RELEASE THE DRIVE FROM PORT A

```

K11

```

(2) 050112 113760 001224 000010 MOV B PORTA,RPCS2(RO) ;SELECT PORT A
(3) 050120 013737 001224 001234 MOV PORTA,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
(3) 050126 012760 000013 000000 MOV #13,RPCS1(RO) ;ISSUE RELEASE THROUGH PORT A
(3) ;VERIFY THAT DRIVE IS SEIZED BY PORT B WHEN RELEASED BY PORT A
(3)
(3) 050134 005037 001250 CLR RELERR ;CLEAR 'RELEASE ERROR' INDICATOR
(3) 050140 012737 111700 001124 MOV #ATA!MOL!PGM!DPR!DRY!VV,$GDDAT ;COMPARISON CONSTANT
(3) 050146 012737 000012 001122 MOV #RPDS1,$BDAOR ;REGISTER ADDRESS INCREMENT
(3) 050154 060037 001122 001122 ADD RO,$BDAOR ;REGISTER BASE ADDRESS FOR TYPEOUT
(4) 050160 113760 001226 000010 MOV B PORTB,RPCS2(RO) ;SELECT PORT B
(4) 050166 013737 001226 001234 MOV PORTB,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
(3) 050174 016037 000012 001164 MOV RPDS1(RO),$TMP0 ;READ STATUS REGISTER FROM PORT B
(4) 050202 113760 001224 000010 MOV B PORTA,RPCS2(RO) ;SELECT PORT A
(4) 050210 013737 001224 001234 MOV PORTA,PTNBR ;MOVE PORT ADDRESS TO LOCAT:JN FOR TYPEOUT
(3) 050216 016037 000012 001126 MOV RPDS1(RO),$BDDAT ;DRIVE STATUS FROM PORT A
(3) 050224 001404 BEQ 66$ ;BR IF STATUS FROM PORT A ZERO
(3) 050226 005737 001164 TST $TMP0 ;IS STATUS FROM PORT B ZERO ?
(3) 050232 001401 BEQ 66$ ;BR IF ZERO
(3) 050234 104031 ERROR 31 ;REPORT DRIVE IN NEUTRAL
(3) 050236 013737 001164 001126 66$: MOV $TMP0,$BDDAT ;CHECK STATUS FROM PORT B
(3) 050244 013737 001226 001234 MOV PORTB,PTNBR ;CHANGE PORT ADDRESS FOR TYPEOUT
(3) 050252 023737 001124 001126 CMP $GDDAT,$BDDAT ;COMPARE WITH CONSTANT
(3) 050260 001401 BEQ 67$ ;BR IF OK
(3) 050262 104027 ERROR 27 ;REPORT REGISTER ERROR
(3) 050264 000240 67$: NOP
(3)
(3) ;RELEASE THE DRIVE FROM PORT B
(3)
(3) 050266 113760 001226 000010 MOV B PORTB,RPCS2(RO) ;SELECT PORT B
(3) 050274 013737 001226 001234 MOV PORTB,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
(3) 050302 012760 000013 000000 MOV #13,RPCS1(RO) ;ISSUE RELEASE THROUGH PORT B
(3)
(3) ;VERIFY THAT THE DRIVE IS IN NEUTRAL
(3)
(3) 050310 005037 001250 CLR RELERR ;CLEAR THE 'RELEASE ERROR' INDICATOR
(3) 050314 012737 000012 001122 MOV #RPDS1,$BDAOR ;FORM THE ADDRESS OF RPDS1 FOR TYPEOUT
(3) 050322 060037 001122 001122 ADD RO,$BDAOR ;ADD THE I/O BASE ADDRESS
(3) 050326 012737 011700 001124 MOV #MOL!PGM!DPR!DRY!VV,$GDDAT ;COMPARISON CONSTANT
(3) 050334 113760 001224 000010 MOV B PORTA,RPCS2(RO) ;SELECT PORT A.
(3) 050342 016037 000012 001170 MOV RPDS1(RO),$TMP2 ;GET THE DRIVE STATUS REGISTER FROM PORT A.
(3) 050350 013737 001170 001164 MOV $TMP2,$TMP0 ;COPY IT INTO '$TMP0'
(3) 050356 042737 100100 001164 BIC #ATA!VV,$TMP0 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
(3) 050364 113760 001226 000010 MOV B PORTB,RPCS2(RO) ;SELECT PORT B.
(3) 050372 016037 000012 001172 MOV RPDS1(RO),$TMP3 ;GET THE DRIVE STATUS REGISTER FROM PORT B.
(3) 050400 013737 001172 001166 MOV $TMP3,$TMP1 ;COPY IT INTO '$TMP1'
(3) 050406 042737 100100 001166 BIC #ATA!VV,$TMP1 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
(3) 050414 023737 001164 001166 CMP $TMP0,$TMP1 ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
(3) 050422 001006 BNE 68$ ;BR IF NOT
(3) 050424 005737 001164 TST $TMP0 ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
(3) 050430 001045 BNE 70$ ;BR IF NOT
(3) 050432 104046 ERROR 46 ;REPORT DRIVE NOT IN NEUTRAL OR NOT SEIZED
(3) 050434 000137 050620 JMP 72$ ;BYPASS THE REST OF THE CHECKS
(3) 050440 013737 001170 001126 68$: MOV $TMP2,$BDDAT ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
(3) 050446 013737 001226 001234 MOV PORTB,PTNBR ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
  
```

```

(3) 050454 113760 001226 000010      MOVB  PORTB,PPCS2(RO) ;SELECT PORT B.
(3) 050462 005737 001164              TST  $TMP0             ;SEE IF STATUS EQ 0 FROM PORT A.
(3) 050466 001414                      BEQ  69$              ;BR IF ZERO
(3) 050470 013737 001224 001234      MOV  PORTA,PTNBR      ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
(3) 050476 013737 001172 001126      MOV  $TMP3,$BDDAT    ;'BAD DATA' FOR ERROR TYPE OUT
(3) 050504 113760 001224 000010      MOVB  PORTA,PPCS2(RO) ;SELECT PORT A.
(3) 050512 005737 001166              TST  $TMP1             ;SEE IF STATUS EQ ZERO FROM PORT B.
(3) 050516 001012                      BNE  70$              ;BR IF NOT
(3) 050520 012737 177777 001250 69$:  MOV  #-1,RELEERR     ;SET 'RELEASE ERROR' INDICATOR
(3) 050526 012760 000011 000000      MOV  #11,PPCS1(RO)   ;CLEAR THE DRIVE
(3) 050534 012760 000013 000000      MOV  #13,PPCS1(RO)   ;RELEASE THE DRIVE
(3) 050542 104026                      ERROR 26              ;TYPE ERROR MESSAGE 26
(3) 050544 013737 001170 001126 70$:  MOV  $TMP2,$BDDAT    ;LOOK FOR BIT FAILURES WHEN RPDS1 READ
(3) 050552 013737 001224 001234      MOV  PORTA,PTNBR     ;CHANGE PORT NUMBER
(3) 050560 023737 001124 001170      CMP  $GDDAT,$TMP2    ;ALL BITS OK ?
(3) 050566 001401                      BEQ  71$              ;BR IF OK FROM PORT A.
(3) 050570 104007                      ERROR 7               ;REPORT ERROR
(3) 050572 013737 001172 001126 71$:  MOV  $TMP3,$BDDAT    ;CHECK RPDS1 FOR BIT FAILURES - FROM PORT B.
(3) 050600 013737 001226 001234      MOV  PORTB,PTNBR    ;CHANGE PORT NUMBER
(3) 050606 023737 001124 001172      CMP  $GDDAT,$TMP3    ;SEE IF READ OK FROM PORT B.
(3) 050614 001401                      BEQ  72$              ;BR IF OK
(3) 050616 104007                      ERROR 7               ;REPORT ERROR
(3) 050620 000240                      NOP                   ;
(1) 050622 000004                      SCOPE                 ;LOOP ?
9252 050624 000137 051162              JMP  TST42            ;GO TO THE NEXT TEST

;CHECK THE REGISTERS ON THE SELECTED PORT

(1) 050630 TST41B:
(3) 050630 005037 001244              CLR  CKERR            ;CLEAR THE 'CHECK ERROR' INDICATOR
(3) 050634 016037 000014 001126      MOV  RPER1(RO), $BDDAT ;GET CONTENTS OF RPER1
(3) 050642 012737 000014 001122      MOV  #RPER1,$B0ADR   ;FORM REGISTER ADDRESS OF ERROR MESSAGE
(3) 050650 060037 001122              ADD  RO,$B0ADR        ;ADD RH11 BASE ADDRESS
(3) 050654 005037 001124              CLR  $GDDAT           ;WHAT REGISTER SHOULD BE
(3) 050660 023737 001124 001126      CMP  $GDDAT,$BDDAT   ;IS THE REGISTER OK ?
(3) 050666 001403                      BEQ  64$              ;BR IF OK
(3) 050670 104006                      ERROR 6               ;TYPE MESSAGE 6
(3) 050672 005137 001244              COM  CKERR            ;SET THE REGISTER COMPARE ERROR INDICATOR
(3) 050676 016037 000000 001126 64$:  MOV  RPCS1(RO), $BDDAT ;GET THE CONTENTS OF RHCS1
(3) 050704 012737 000000 001122      MOV  #RPCS1,$B0ADR   ;FORM ADDRESS OF REGISTER
(3) 050712 060037 001122              ADD  RO,$B0ADR        ;ADDRESS BASE
(3) 050716 032737 020000 001126      BIT  #MCPE,$BDDAT    ;IS 'MCPE' SET ?
(3) 050724 001404                      BEQ  65$              ;BR IF NOT
(3) 050726 104011                      ERROR 11              ;REPORT THE ERROR
(3) 050730 012760 040000 000000      MOV  #TRE,PPCS1(RO) ;CLEAR 'MCPE'
(3) 050736 000240                      NOP                   ;
(3) 050740 005037 001244              CLR  CKERR            ;CLEAR THE 'CHECK ERROR' INDICATOR
(3) 050744 016037 000040 001126      MOV  RPER2(RO), $BDDAT ;GET CONTENTS OF RPER2
(3) 050752 012737 000040 001122      MOV  #RPER2,$B0ADR   ;FORM REGISTER ADDRESS OF ERROR MESSAGE
(3) 050760 060037 001122              ADD  RO,$B0ADR        ;ADD RH11 BASE ADDRESS
(3) 050764 005037 001124              CLR  $GDDAT           ;WHAT REGISTER SHOULD BE
(3) 050770 023737 001124 001126      CMP  $GDDAT,$BDDAT   ;IS THE REGISTER OK ?
(3) 050776 001403                      BEQ  66$              ;BR IF OK
(3) 051000 104006                      ERROR 6               ;TYPE MESSAGE 6
(3) 051002 005137 001244              COM  CKERR            ;SET THE REGISTER COMPARE ERROR INDICATOR
(3) 051006 016037 000000 001126 66$:  MOV  RPCS1(RO), $BDDAT ;GET THE CONTENTS OF RHCS1
    
```

```

(3) 051014 012737 000000 001122 MOV #RPCS1,$B0ADR ;FORM ADDRESS OF REGISTER
(3) 051022 060037 001122 ADD R0,$B0ADR ;ADDRESS BASE
(3) 051026 032737 020000 001126 BIT #MCPE,$BDDAT ;IS 'MCPE' SET ?
(3) 051034 001404 BEQ 67$ ;BR IF NOT
(3) 051036 104011 ERROR 11 ;REPORT THE ERROR
(3) 051040 012760 040000 000000 MOV #TRE,RPCS1(R0) ;CLEAR 'MCPE'
(3) 051046 000240 67$: NOP
(3) 051050 005037 001244 CLR CKERR ;CLEAR THE 'CHECK ERROR' INDICATOR
(3) 051054 016037 000042 001126 MOV RPER3(R0),$BDDAT ;GET CONTENTS OF RPER3
(3) 051062 012737 000042 001122 MOV #RPER3,$B0ADR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
(3) 051070 060037 001122 ADD R0,$B0ADR ;ADD RH11 BASE ADDRESS
(3) 051074 005037 001124 CLR $GDDAT ;WHAT REGISTER SHOULD BE
(3) 051100 023737 001124 001126 CMP $GDDAT,$BDDAT ;IS THE REGISTER OK ?
(3) 051106 001403 BEQ 68$ ;BR IF OK
(3) 051110 104006 ERROR 6 ;TYPE MESSAGE 6
(3) 051112 005137 001244 COM CKERR ;SET THE REGISTER COMPARE ERROR INDICATOR
(3) 051116 016037 000000 001126 68$: MOV RPCS1(R0),$BDDAT ;GET THE CONTENTS OF RHCS1
(3) 051124 012737 000000 001122 MOV #RPCS1,$B0ADR ;FORM ADDRESS OF REGISTER
(3) 051132 060037 001122 ADD R0,$B0ADR ;ADDRESS BASE
(3) 051136 032737 020000 001126 BIT #MCPE,$BDDAT ;IS 'MCPE' SET ?
(3) 051144 001404 BEQ 69$ ;BR IF NOT
(3) 051146 104011 ERROR 11 ;REPORT THE ERROR
(3) 051150 012760 040000 000000 MOV #TRE,RPCS1(R0) ;CLEAR 'MCPE'
(3) 051156 000240 69$: NOP
(1) 051160 000207 RTS PC ;RETURN

```

9277
9278

```

*****
*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.
*
*****

```

```

(2) 051162 TST42: TST KYBCTL ;PERFORMING ONLY SINGLE TESTS ?
(3) 051162 BEQ 2$ ;BR IF NOT
(3) 051166 BPL 1$ ;BR IF JUST ENTERED TEST
(3) 051170 JMP EXEC ;RETURN & GET NEXT TEST NUMBER
(3) 051172

```

```

(3) 051176 012737 177777 001274 15: MOV #1,KYBCTL ;SET SINGLE TEST INDICATOR
(3) 051204 112737 000042 001102 25: MOV #42,$TSTNM ;TEST NUMBER
(3) 051212 012737 051234 001106 MOV #TEST42,$LPADR ;LOAD LOOP ON TEST ADDRESS
(3) 051220 012737 051234 001110 MOV #TEST42,$LPERR ;LOAD LOOP ON ERROR ADDRESS
(1) 051226 012737 007640 001176 MOV #4000,$TIMES ;DO 4000. ITERATIONS
9279 051234 012706 001100 TEST42: MOV #STACK,$P ;LOAD THE STACK POINTER
9280
(2) ;CLEAR ATTENTION BITS FOR BOTH PORTS
(2) 051240 113760 001224 000010 MOVB PORTA,RPCS2(RO) ;SELECT PORT #A
(2) 051246 005060 000012 CLR RPDS1(RO) ;SEIZE THE DPIPE
(2) 051252 012760 000011 000C00 MOV #11,RPCS1(RO) ;ISSUE DRIVE CLEAR
(2) 051260 012760 000013 000000 MOV #13,RPCS1(RO) ;RELEASE THE DRIVE
(2) 051266 113760 001226 000010 MOVB PORTB,RPCS2(RO) ;SELECT PORT #B
(2) 051274 005060 000012 CLR RPDS1(RO) ;SEIZE THE DRIVE THROUGH PORT 'B'
(2) 051300 012760 000011 000000 MOV #11,RPCS1(RO) ;ISSUE DRIVE CLEAR
(2) 051306 012760 000013 000000 MOV #13,RPCS1(RO) ;RELEASE THE DRIVE
(2) ;SEIZE THE DRIVE THROUGH PORT B
(2) 051314 113760 001226 000010 MOVB PORTB,RPCS2(RO) ;SELECT PORT B
(2) 051322 013737 001226 001236 MOV PORTB,SEIZPT ;STORE SEIZING PORT'S ADDRESS
(2) 051330 005060 000012 CLR RPDS1(RO) ;WRITE RPDS1
(2) 051334 013737 001224 001240 MOV PORTA,OPPRT ;'OPPOSITE' PORT ADDRESS
(2) 051342 012760 177777 000014 MOV #-1,RPER1(RO) ;LOAD 1'S INTO RPER1 THROUGH PORT B
(2) 051350 012760 177777 000040 MOV #-1,RPER2(RO) ;LOAD 1'S INTO RPER2 THROUGH PORT B
(2) 051356 012760 177777 000042 MOV #-1,RPER3(RO) ;LOAD 1'S INTO RPER3 THROUGH PORT B
(2) 051364 113760 001224 000010 MOVB PORTA,RPCS2(RO) ;SELECT PORT A
(2) 051372 013737 001224 001234 MOV PORTA,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
(1) 051400 004737 052236 JSR PC,TST42B ;CHECK THE REGISTERS THROUGH PORT A
(2) 051404 113760 001226 000010 MOVB PORTB,RPCS2(RO) ;SELECT PORT B
(2) 051412 013737 001226 001234 MOV PORTB,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
(2) 051420 005060 000042 CLR RPER3(RO) ;CLEAR RPER3 ON PORT B
(2) 051424 005060 000040 CLR RPER2(RO) ;CLEAR RPER2 ON PORT B
(2) 051430 005060 000014 CLR RPER1(RO) ;CLEAR RPER1 ON PORT B
(1) 051434 013760 001232 000016 MOV ASR1,RPAS(RO) ;CLEAR THE ATTENTION BIT FOR PORT B
(2) 051442 113760 001224 000010 MOVB PORTA,RPCS2(RO) ;SELECT PORT A
(2) 051450 013737 001224 001234 MOV PORTA,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
(2) 051456 012760 177777 000014 MOV #-1,RPER1(RO) ;LOAD 1'S INTO RPER1 THROUGH PORT A
(2) 051464 012760 177777 000040 MOV #-1,RPER2(RO) ;LOAD 1'S INTO RPER2 THROUGH PORT A
(2) 051472 012760 177777 000042 MOV #-1,RPER3(RO) ;LOAD 1'S INTO RPER3 THROUGH PORT A
(2) 051500 113760 001226 000010 MOVB PORTB,RPCS2(RO) ;SELECT PORT B
(2) 051506 013737 001226 001234 MOV PORTB,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
(1) 051514 004737 052236 JSR PC,TST42B ;CHECK THE REGISTERS THROUGH PORT B
(2) ;RELEASE THE DRIVE FROM PORT B
(2) 051520 113760 001226 000010 MOVB PORTB,RPCS2(RO) ;SELECT PORT B
(3) 051526 013737 001226 001234 MOV PORTB,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
(2) 051534 012760 000013 000000 MOV #13,RPCS1(RO) ;ISSUE RELEASE THROUGH PORT B
(3) ;VERIFY THAT DRIVE IS SEIZED BY PORT A WHEN RELEASED BY PORT B
(3) 051542 005037 001250 CLR RELERR ;CLEAR 'RELEASE ERROR' INDICATOR
(3) 051546 012737 111700 001124 MOV #ATA!MOL!PGM!DPR!DRY!VV,$GDDAT ;COMPARISON CONSTANT
(3) 051554 012737 000012 001122 MOV #RPDS1,$BDADR ;REGISTER ADDRESS INCREMENT
    
```



```

(3) 051562 060037 001122      ADD      RO,$B0ADR      ;REGISTER BASE ADDRESS FOR TYPEOUT
(4) 051566 113760 001224 000010  MOVB    PORTA,RPCS2(RO) ;SELECT PORT A
(4) 051574 013737 001224 001234  MOV     PORTA,PTNBR     ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
(3) 051602 016037 000012 001164  MOV     RPDS1(RO),STMP0 ;READ STATUS REGISTER FROM PORT A
(4) 051610 113760 001226 000010  MOVB    PORTB,RPCS2(RO) ;SELECT PORT B
(4) 051616 013737 001226 001234  MOV     PORTB,PTNBR     ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
(3) 051624 016037 000012 001126  MOV     RPDS1(RO),SBDDAT ;DRIVE STATUS FROM PORT B
(3) 051632 001404      BEQ     66$             ;BR IF STATUS FROM PORT B ZERO
(3) 051634 005737 001164      TST     STMP0           ;IS STATUS FROM PORT A ZERO ?
(3) 051640 001401      BEQ     66$             ;BR IF ZERO
(3) 051642 104031      ERROR   31             ;REPORT DRIVE IN NEUTRAL
(3) 051644 013737 001164 001126 66$:  MOV     STMP0,$BDDAT    ;CHECK STATUS FROM PORT A
(3) 051652 013737 001224 001234  MOV     PORTA,PTNBR     ;CHANGE PORT ADDRESS FOR TYPEOUT
(3) 051660 023737 001124 001126  CMP     $GDDAT,$BDDAT   ;COMPARE WITH CONSTANT
(3) 051666 001401      BEQ     67$             ;BR IF OK
(3) 051670 104027      ERROR   27             ;REPORT REGISTER ERROR
(3) 051672 000240      NOP

(2)
(2)
(2)
(3) 051674 113760 001224 000010  MOVB    PORTA,RPCS2(RO) ;SELECT PORT A
(3) 051702 013737 001224 001234  MOV     PORTA,PTNBR     ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
(2) 051710 012760 000013 000000  MOV     #13,RPDS1(RO)   ;ISSUE RELEASE THROUGH PORT A

(3)
(3)
(3)
(3)
(3) 051716 005037 001250      CLR     RELERR         ;CLEAR THE 'RELEASE ERROR' INDICATOR
(3) 051722 012737 000012 001122  MOV     #RPDS1,$B0ADR   ;FORM THE ADDRESS OF RPDS1 FOR TYPEOUT
(3) 051730 060037 001122      ADD     RO,$B0ADR      ;ADD THE I/O BASE ADDRESS
(3) 051734 012737 011700 001124  MOV     #MOL:PGM:DPR:DRY:VV,$GDDAT ;COMPARISON CONSTANT
(3) 051742 113760 001224 000010  MOVB    PORTA,RPCS2(RO) ;SELECT PORT A.
(3) 051750 016037 000012 001170  MOV     RPDS1(RO),STMP2 ;GET THE DRIVE STATUS REGISTER FROM PORT A.
(3) 051756 013737 001170 001164  MOV     STMP2,STMP0     ;COPY IT INTO 'STMP0'
(3) 051764 042737 100100 001164  BIC     #ATA!VV,STMP0   ;CLEAR PORT DEPENDENT BITS FROM THE COPY
(3) 051772 113760 001226 000010  MOVB    PORTB,RPCS2(RO) ;SELECT PORT B.
(3) 052000 016037 000012 001172  MOV     RPDS1(RO),STMP3 ;GET THE DRIVE STATUS REGISTER FROM PORT B.
(3) 052006 013737 001172 001166  MOV     STMP3,STMP1    ;COPY IT INTO 'STMP1'
(3) 052014 042737 100100 001166  BIC     #ATA!VV,STMP1   ;CLEAR PORT DEPENDENT BITS FROM THE COPY
(3) 052022 023737 001164 001166  CMP     STMP0,STMP1    ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
(3) 052030 001006      BNE     69$            ;BR IF NOT
(3) 052032 005737 001164      TST     STMP0           ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
(3) 052036 001045      BNE     70$            ;BR IF NOT
(3) 052040 104046      ERROR   46             ;REPORT DRIVE NOT IN NEUTRAL OR NOT SEIZED
(3) 052042 000137 052226      JMP     72$            ;BYPASS THE REST OF THE CHECKS
(3) 052046 013737 001170 001126 68$:  MOV     STMP2,$BDDAT    ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
(3) 052054 013737 001226 001234  MOV     PORTB,PTNBR     ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
(3) 052062 113760 001226 000010  MOVB    PORTB,RPCS2(RO) ;SELECT PORT B.
(3) 052070 005737 001164      TST     STMP0           ;SEE IF STATUS EQ 0 FROM PORT A.
(3) 052074 001414      BEQ     69$            ;BR IF ZERO
(3) 052076 013737 001224 001234  MOV     PORTA,PTNBR     ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
(3) 052104 013737 001172 001126  MOV     STMP3,$BDDAT    ;'BAD DATA' FOR ERROR TYPE OUT
(3) 052112 113760 001224 000010  MOVB    PORTA,RPCS2(RO) ;SELECT PORT A.
(3) 052120 005737 001166      TST     STMP1           ;SEE IF STATUS EQ ZERO FROM PORT B.
(3) 052124 001012      BNE     70$            ;BR IF NOT
(3) 052126 012737 177777 001250 69$:  MOV     #-1,RELERR     ;SET 'RELEASE ERROR' INDICATOR
(3) 052134 012760 000011 000000  MOV     #11,RPDS1(RO)  ;CLEAR THE DRIVE
    
```

```

(3) 052142 012760 000013 000000      MOV      #13,RPCS1(RO) ;RELEASE THE DRIVE
(3) 052150 104026                ERROR    26           ;TYPE ERROR MESSAGE 26
(3) 052152 013737 001170 001126 70$:  MOV      $TMP2,$BDDAT ;LOOK FOR BIT FAILURES WHEN RPOSI READ
(3) 052160 013737 001224 001234      MOV      PORTA,PTNBR  ;CHANGE PORT NUMBER
(3) 052166 023737 001124 001170      CMP      $GDDAT,$TMP2 ;ALL BITS OK ?
(3) 052174 001401                BEQ     71$         ;BR IF OK FROM PORT A.
(3) 052176 104007                ERROR    7           ;REPORT ERROR
(3) 052200 013737 001172 001126 71$:  MOV      $TMP3,$BDDAT ;CHECK RPOSI FOR BIT FAILURES - FROM PORT B.
(3) 052206 013737 001226 001234      MOV      PORTB,PTNBR ;CHANGE PORT NUMBER
(3) 052214 023737 001124 001172      CMP      $GDDAT,$TMP3 ;SEE IF READ OK FROM PORT B.
(3) 052222 001401                BEQ     72$         ;BR IF OK
(3) 052224 104007                ERROR    7           ;REPORT ERROR
(3) 052226 000240                NOP
(1) 052230 000004                SCOPE
9281 052232 000137 052570      JMP     TST43        ;LOOP ?
(1)                                     ;GO TO THE NEXT TEST
(1)                                     ;CHECK THE REGISTERS ON THE SELECTED PORT
(1) TST42B:
(3) 052236 005037 001244                CLR     CKERR        ;CLEAR THE 'CHECK ERROR' INDICATOR
(3) 052242 016037 000014 001126      MOV     RPER1(RO), $BDDAT ;GET CONTENTS OF RPER1
(3) 052250 012737 000014 001122      MOV     #RPER1,$B0ADR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
(3) 052256 060037 001122                ADD     RO,$B0ADR    ;ADD RH11 BASE ADDRESS
(3) 052262 005037 001124                CLR     $GDDAT      ;WHAT REGISTER SHOULD BE
(3) 052266 023737 001124 001126      CMP     $GDDAT,$BDDAT ;IS THE REGISTER OK ?
(3) 052274 001403                BEQ     64$         ;BR IF OK
(3) 052276 104006                ERROR    6           ;TYPE MESSAGE 6
(3) 052300 005137 001244                COM     CKERR        ;SET THE REGISTER COMPARE ERROR INDICATOR
(3) 052304 016037 000000 001126 64$:  MOV     RPCS1(RO), $BDDAT ;GET THE CONTENTS OF RHCS1
(3) 052312 012737 000000 001122      MOV     #RPCS1,$B0ADR ;FORM ADDRESS OF REGISTER
(3) 052320 060037 001122                ADD     RO,$B0ADR    ;ADDRESS BASE
(3) 052324 032737 020000 001126      BIT     #MCPE,$BDDAT ;IS 'MCPE' SET ?
(3) 052332 001404                BEQ     65$         ;BR IF NOT
(3) 052334 104011                ERROR    11          ;REPORT THE ERROR
(3) 052336 012760 040000 000000 65$:  MOV     #TRE,RPCS1(RO) ;CLEAR 'MCPE'
(3) 052344 000240                NOP
(3) 052346 005037 001244                CLR     CKERR        ;CLEAR THE 'CHECK ERROR' INDICATOR
(3) 052352 016037 000040 001126      MOV     RPER2(RO), $BDDAT ;GET CONTENTS OF RPER2
(3) 052360 012737 000040 001122      MOV     #RPER2,$B0ADR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
(3) 052366 060037 001122                ADD     RO,$B0ADR    ;ADD RH11 BASE ADDRESS
(3) 052372 005037 001124                CLR     $GDDAT      ;WHAT REGISTER SHOULD BE
(3) 052376 023737 001124 001126      CMP     $GDDAT,$BDDAT ;IS THE REGISTER OK ?
(3) 052404 001403                BEQ     66$         ;BR IF OK
(3) 052406 104006                ERROR    6           ;TYPE MESSAGE 6
(3) 052410 005137 001244                COM     CKERR        ;SET THE REGISTER COMPARE ERROR INDICATOR
(3) 052414 016037 000000 001126 66$:  MOV     RPCS1(RO), $BDDAT ;GET THE CONTENTS OF RHCS1
(3) 052422 012737 000000 001122      MOV     #RPCS1,$B0ADR ;FORM ADDRESS OF REGISTER
(3) 052430 060037 001122                ADD     RO,$B0ADR    ;ADDRESS BASE
(3) 052434 032737 020000 001126      BIT     #MCPE,$BDDAT ;IS 'MCPE' SET ?
(3) 052442 001404                BEQ     67$         ;BR IF NOT
(3) 052444 104011                ERROR    11          ;REPORT THE ERROR
(3) 052446 012760 040000 000000 67$:  MOV     #TRE,RPCS1(RO) ;CLEAR 'MCPE'
(3) 052454 000240                NOP
(3) 052456 005037 001244                CLR     CKERR        ;CLEAR THE 'CHECK ERROR' INDICATOR
(3) 052462 016037 000042 001126      MOV     RPER3(RO), $BDDAT ;GET CONTENTS OF RPER3
(3) 052470 012737 000042 001122      MOV     #RPER3,$B0ADR ;FORM REGISTER ADDRESS OF ERROR MESSAGE

```

```

(3) 052476 060037 001122      ADD    RO,$BDADR      ;ADD RH11 BASE ADDRESS
(3) 052502 005037 001124      CLR    $GDDAT        ;WHAT REGISTER SHOULD BE
(3) 052506 023737 001124 001126  CMP    $GDDAT,$BDDAT ;IS THE REGISTER OK ?
(3) 052514 001403      BEQ    68$           ;BR IF OK
(3) 052516 104006      ERROR  6           ;TYPE MESSAGE 6
(3) 052520 005137 001244      COM    CKERR        ;SET THE REGISTER COMPARE ERROR INDICATOR
(3) 052524 016037 000000 001126 62$:  MOV    RPCS1(RO),$BDDAT ;GET THE CONTENTS OF RHCS1
(3) 052532 012737 000000 001122  MOV    #RPCS1,$B0ADR  ;FORM ADDRESS OF REGISTER
(3) 052540 060037 001122      ADC    RO,$BDADR    ;ADDRESS BASE
(3) 052544 032737 020000 001126  BIT    #MCPE,$BDDAT  ;IS 'MCPE' SET ?
(3) 052552 001404      BEQ    69$           ;BR IF NOT
(3) 052554 104011      ERROR  11          ;REPORT THE ERROR
(3) 052556 012760 040000 000      MOV    #TRE,RPCS1(RO);CLEAR 'MCPE'
(3) 052564 000240      NOP
(1) 052566 000207      RTS    PC           ;RETURN

```

9282
9295
9296

```

*****
*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.
*****

```

(2)
(3)
(3)
(3)
(3)
(3)
(3)
(3)
(3)
(3)
(1)
9297
9330

```

*****
*TEST43:
TST    KYBCTL      ;PERFORMING ONLY SINGLE TESTS ?
BEQ    2$          ;BR IF NOT
BPL    1$          ;BR IF JUST ENTERED TEST
JMP    EXEC        ;RETURN & GET NEXT TEST NUMBER
1$:    MOV    #-1,KYBCTL ;SET SINGLE TEST INDICATOR
2$:    MOV    #43,$STNM  ;TEST NUMBER
      MOV    #TEST43,$LPADR ;LOAD LOOP ON TEST ADDRESS
      MOV    #TEST43,$LPERR ;LOAD LOOP ON ERROR ADDRESS
      MOV    #25,$TIMES  ;DO 25 ITERATIONS
TEST43: MOV    #STACK,SP ;LOAD THE STACK POINTER

```

;CLEAR ATTENTION BITS FOR BOTH PORTS

```

(2) 052646 113760 001224 000010  MOV    PORTA,RPCS2(RO) ;SELECT PORT #A
(2) 052654 005060 000012 000000  CLR    RPDS1(RO)      ;SEIZE THE DRIVE
(2) 052660 012760 000011 000000  MOV    #11,RPCS1(RO)  ;ISSUE DRIVE CLEAR
(2) 052666 012760 000013 000000  MOV    #13,RPCS1(RO)  ;RELEASE THE DRIVE
(2) 052674 113760 001226 000010  MOV    PORTB,RPCS2(RO);SELECT PORT #B
(2) 052702 005060 000012 000000  CLR    RPDS1(RO)      ;SEIZE THE DRIVE THROUGH PORT 'B'
(2) 052706 012760 000011 000000  MOV    #11,RPCS1(RO)  ;ISSUE DRIVE CLEAR
(2) 052714 012760 000013 000000  MOV    #13,RPCS1(RO)  ;RELEASE THE DRIVE
(1) 052722 113760 001224 000010  MOV    PORTA,RPCS2(RO);SELECT PORT A
(1) 052730 012760 177777 000014  MOV    #-1,RPER1(RO) ;SET ERRORS TO FORCE ATTN BIT ON PORT A
(1) 052736 005060 000014 000000  CLR    RPER1(RO)     ;CLEAR THE ERRORS
(1) 052742 113760 001226 000010  MOV    PORTB,RPCS2(RO);SELECT PORT B

```

```

(1) 052750 005760 000012 15: TST RPDS1(RO) ;WAIT FOR DRIVE TO RETURN TO NEUTRAL
(1) 052754 001775 BEQ 15 ;BR IF STILL SEIZED BY PORT A
(1) 052756 012737 000016 001122 MOV #RPAS,$B0ADR ;FORM ADDRESS OF ATTN REG IF ERROR
(1) 052764 060037 001122 ADD RO,$B0ADR ;ADD THE ADDRESS BASE
(1) 052770 013737 001232 001124 MOV ASR1,$GDDAT ;GOOD DATA FOR ERROR MESSAGE
(1) 052776 013737 001232 001166 MOV ASR1,$TMP1 ;MAKE DATA COMPARE MASK
(1) 053004 005137 001166 COM $TMP1 ;COMPLEMENT IT
(1) 053010 012737 053044 001110 MOV #25,$LPERR ;LOAD LOOP ON ERROR ADDRESS
(2) 053016 113760 001226 000010 MOVB PORTB,RPCS2(RO) ;SELECT PORT B
(2) 053024 013737 001226 001234 MOV PORTB,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
(1) 053032 013737 001226 001236 MOV PORTB,SEIZPT ;'SEIZED' PORT ADDRESS
(1) 053040 005060 000012 CLR RPDS1(RO) ;SEIZE THE DRIVE THROUGH PORT B
(1) 053044 016037 000016 001126 25: MOV RPAS(RO),$B0DAT ;GET THE CONTENTS OF THE ATTENTION REG
(1) 053052 013737 001126 001164 MOV $B0DAT,$TMP0 ;PUT CONTENTS INTO WORKING LOCATION
(1) 053060 043737 001166 001164 BIC $TMP1,$TMP0 ;CLEAR OTHER BITS
(1) 053066 023737 001124 001164 CMP $GDDAT,$TMP0 ;SEE IF ATTN BIT FOR DRIVE SET
(1) 053074 001401 BEQ 35 ;BR IF SET
(1) 053076 104053 ERROR 53 ;REPORT THE ERROR
(1) 053100 35:

;RELEASE THE DRIVE FROM PORT B
(3) 053100 113760 001226 000010 MOVB PORTB,RPCS2(RO) ;SELECT PORT B
(3) 053106 013737 001226 001234 MOV PORTB,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
(3) 053114 012760 000013 000000 MOV #13,RPDS1(RO) ;ISSUE RELEASE THROUGH PORT B

;VERIFY THAT THE DRIVE IS IN NEUTRAL
(3) 053122 005037 001250 CLR RELERR ;CLEAR THE 'RELEASE ERROR' INDICATOR
(3) 053126 012737 000012 001122 MOV #RPDS1,$B0ADR ;FORM THE ADDRESS OF RPDS1 FOR TYPEOUT
(3) 053134 060037 001122 ADD RO,$B0ADR ;ADD THE I/O BASE ADDRESS
(3) 053140 012737 011700 001124 MOV #MOL:PCM:DPR:DRY:VV,$GDDAT ;COMPARISON CONSTANT
(3) 053146 113760 001224 000010 MOVB PORTA,RPCS2(RO) ;SELECT PORT A.
(3) 053154 016037 000012 001170 MOV RPDS1(RO),$TMP2 ;GET THE DRIVE STATUS REGISTER FROM PORT A.
(3) 053162 013737 001170 001164 MOV $TMP2,$TMP0 ;COPY IT INTO '$TMP0'
(3) 053170 042737 100100 001164 BIC #ATA:VV,$TMP0 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
(3) 053176 113760 001226 000010 MOVB PORTB,RPCS2(RO) ;SELECT PORT B.
(3) 053204 016037 000012 001172 MOV RPDS1(RO),$TMP3 ;GET THE DRIVE STATUS REGISTER FROM PORT B.
(3) 053212 013737 001172 001166 MOV $TMP3,$TMP1 ;COPY IT INTO '$TMP1'
(3) 053220 042737 100100 001166 BIC #ATA:VV,$TMP1 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
(3) 053226 023737 001164 001166 CMP $TMP0,$TMP1 ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
(3) 053234 001006 BNE 64$ ;BR IF NOT
(3) 053236 005737 001164 TST $TMP0 ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
(3) 053242 001045 BNE 66$ ;BR IF NOT
(3) 053244 104046 ERROR 46 ;REPORT DRIVE NOT IN NEUTRAL OR NOT SEIZED
(3) 053246 000137 053446 JMP 68$ ;BYPASS THE REST OF THE CHECKS
(3) 053252 013737 001170 001126 64$: MOV $TMP2,$B0DAT ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
(3) 053260 013737 001226 001234 MOV PORTB,PTNBR ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
(3) 053266 113760 001226 000010 MOVB PORTB,RPCS2(RO) ;SELECT PORT B.
(3) 053274 005737 001164 TST $TMP0 ;SEE IF STATUS EQ 0 FROM PORT A.
(3) 053300 001414 BEQ 65$ ;BR IF ZERO
(3) 053302 013737 001224 001234 MOV PORTA,PTNBR ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
(3) 053310 013737 001172 001126 MOV $TMP3,$B0DAT ;'BAD DATA' FOR ERROR TYPE OUT
(3) 053316 113760 001224 000010 MOVB PORTA,RPCS2(RO) ;SELECT PORT A.
(3) 053324 005737 001166 TST $TMP1 ;SEE IF STATUS EQ ZERO FROM PORT B.
(3) 053330 001012 BNE 66$ ;BR IF NOT

```

```

(3) 053332 012737 177777 001250 65$: MOV #-1,RELEA ;SET 'RELEASE ERROR' INDICATOR
(3) 053340 012760 000011 000000 MOV #11,RPDS1(RO) ;CLEAR THE DRIVE
(3) 053346 012760 000013 000000 MOV #13,RPDS1(RO) ;RELEASE THE DRIVE
(3) 053354 104026 ERROR 26 ;TYPE ERROR MESSAGE 26
(3) 053356 013737 001170 001126 66$: MOV $TMP2,$BDDAT ;LOOK FOR BIT FAILURES WHEN RPDS1 READ
(3) 053364 013737 001224 001234 MOV PORTA,PTNBR ;CHANGE PORT NUMBER
(3) 053372 042737 100000 001170 BIC #ATH,$TMP2 ;DON'T CHECK THE ATTN BIT
(3) 053400 023737 001124 001170 CMP $GDDAT,$TMP2 ;ALL BITS OK?
(3) 053406 001401 BEQ 67$ ;BR IF OK FROM PORT A.
(3) 053410 104007 ERROR 7 ;REPORT ERROR
(3) 053412 013737 001172 001126 67$: MOV $TMP3,$BDDAT ;CHECK RPDS1 FOR BIT FAILURES - FROM PORT B.
(3) 053420 013737 001226 001234 MOV PORTB,PTNBR ;CHANGE PORT NUMBER
(3) 053426 042737 100000 001172 BIC #ATA,$TMP3 ;DON'T CHECK THE ATTN BIT
(3) 053434 023737 001124 001172 CMP $GDDAT,$TMP3 ;SEE IF READ OK FROM PORT B.
(3) 053442 001401 BEQ 68$ ;BR IF OK
(3) 053444 104007 ERROR 7 ;REPORT ERROR
(3) 053446 000240 68$: NOP
(3) 053450 000004 SCOPE ;LOOP ?

```

```

*****
*TEST 44 TEST PORT 'B' ALTERNATE ATTENTION BIT PATH
*
*VERIFY THAT THE ALTERNATE ATTENTION REGISTER READ PATH IS OPERATIONAL.
*
* A. SET THE ATTENTION BIT FOR PORT 'B'.
*
* B. SEIZE THE DRIVE THROUGH PORT 'A' BY WRITING 0'S INTO RPDS1.
*
* C. READ THE ATTENTION REGISTER & VERIFY THAT THE ATTENTION BIT
* FOR THE DRIVE IS SET.
*****

```

```

(3) 053452 005737 001274 ST44: TST KYBCTL ;PERFORMING ONLY SINGLE TESTS ?
(3) 053456 001406 BEQ 2$ ;BR IF NOT
(3) 053460 100002 BPL 1$ ;BR IF JUST ENTERED TEST
(3) 053462 000137 002622 JMP EXEC ;RETURN & GET NEXT TEST NUMBER
(3) 053466 012737 177777 001274 1$: MOV #-1,KYBCTL ;SET SINGLE TEST INDICATOR
(3) 053474 112737 000044 001102 2$: MOV #44,$STNM ;TEST NUMBER
(3) 053502 012737 053524 001106 MOV #TEST44,$LPADR ;LOAD LOOP ON TEST ADDRESS
(3) 053510 012737 053524 001110 MOV #TEST44,$LPERR ;LOAD LOOP ON ERROR ADDRESS
(3) 053516 012737 000031 001176 MOV #25,$TIMES ;DO 25 ITERATIONS
(3) 053524 012706 001100 TEST44: MOV #STACK,SP ;LOAD THE STACK POINTER

```

;CLEAR ATTENTION BITS FOR BOTH PORTS

```

(3) 053530 113760 001224 000010 MOV#B PORTA,RPDS2(RO) ;SELECT PORT #A
(3) 053536 005750 000012 CLR RPDS1(RO) ;SEIZE THE DRIVE
(3) 053542 012760 000011 000000 MOV #11,RPDS1(RO) ;ISSUE DRIVE CLEAR
(3) 053550 012760 000013 000000 MOV #13,RPDS1(RO) ;RELEASE THE DRIVE
(3) 053556 113760 001226 000010 MOV#B PORTB,RPDS2(RO) ;SELECT PORT #B
(3) 053564 005060 000012 CLR RPDS1(RO) ;SEIZE THE DRIVE THROUGH PORT 'B'
(3) 053570 012760 000011 000000 MOV #11,RPDS1(RO) ;ISSUE DRIVE CLEAR
(3) 053576 012760 000013 000000 MOV #13,RPDS1(RO) ;RELEASE THE DRIVE
(3) 053604 113760 001226 000010 MOV#B PORTB,RPDS2(RO) ;SELECT PORT B
(3) 053612 012760 177777 000014 MOV #-1,RPERR(RO) ;SET ERRORS TO FORCE ATTN BIT ON PORT B

```

```

1 053620 005060 000014 CLR RPER1(RO) ;CLEAR THE ERRORS
1 053624 113760 001224 000010 MOVB PORTA,RPCS2(RO) ;SELECT PORT A
1 053632 005760 000012 15: TST RPDS1(RO) ;WAIT FOR DRIVE TO RETURN TO NEUTRAL
1 053636 001775 BEQ 15 ;BR IF STILL SEIZED BY PORT B
1 053640 012737 000016 001122 MOV #RPAS,$B0ADR ;FORM ADDRESS OF ATTN REG IF ERROR
1 053646 060037 001122 ADD RO,$B0ADR ;ADD THE ADDRESS BASE
1 053652 013737 001232 001124 MOV ASR1,$GDDAT ;GOOD DATA FOR ERROR MESSAGE
1 053660 013737 001232 001166 MOV ASR1,$TMP1 ;MAKE DATA COMPARE MASK
1 053666 005137 001166 COM $TMP1 ;COMPLEMENT IT
1 053672 012737 053726 001110 MOV #25,$LPERR ;LOAD LOOP ON ERROR ADDRESS
1 053700 113760 001224 000010 MOVB PORTA,RPCS2(RO) ;SELECT PORT A
1 053706 013737 001224 001234 MOV PORTA,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
1 053714 013737 001224 001236 MOV PORTA,SEIZPT ;'SEIZED' PORT ADDRESS
1 053722 005060 000012 CLR RPDS1(RO) ;SEIZE THE DRIVE THROUGH PORT A
1 053726 016037 000016 25: MOV RPAS(RO),$BDDAT ;GET THE CONTENTS OF THE ATTENTION REG
1 053734 013737 001126 001164 MOV $BDDAT,$TMP0 ;PUT CONTENTS INTO WORKING LOCATION
1 053742 043737 001166 001164 BIC $TMP1,$TMP0 ;CLEAR OTHER BITS
1 053750 023737 001124 001164 CMP $GDDAT,$TMP0 ;SEE IF ATTN BIT FOR DRIVE SET
1 053756 001401 BEQ 35 ;BR IF SET
1 053760 104053 ERROR 53 ;REPORT THE ERROR
1 053762 113760 001224 000010 ;RELEASE THE DRIVE FROM PORT A
1 053770 013737 001224 001234 MOVB PORTA,RPCS2(RO) ;SELECT PORT A
1 053776 012760 000013 000000 MOV PORTA,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
1 MOV #13,RPCS1(RO) ;ISSUE RELEASE THROUGH PORT A
1 ;VERIFY THAT THE DRIVE IS IN NEUTRAL
1 054004 005037 001250 CLR RELERR ;CLEAR THE 'RELEASE ERROR' INDICATOR
1 054010 012737 000012 001122 MOV #RPDS1,$B0ADR ;FORM THE ADDRESS OF RPDS1 FOR TYPEOUT
1 054016 060037 001122 ADD RO,$B0ADR ;ADD THE I/O BASE ADDRESS
1 054022 012737 011700 001124 MOV #MOL!PGM!DPR!DRY!VV,$GDDAT ;COMPARISON CONSTANT
1 054030 113760 001224 000010 MOVB PORTA,RPCS2(RO) ;SELECT PORT A
1 054036 016037 000012 001170 MOV RPDS1(RO),$TMP2 ;GET THE DRIVE STATUS REGISTER FROM PORT A
1 054044 013737 001170 001164 MOV $TMP2,$TMP0 ;COPY IT INTO '$TMP0'
1 054052 042737 100100 001164 BIC #ATA!VV,$TMP0 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
1 054060 113760 001226 000010 MOVB PORTB,RPCS2(RO) ;SELECT PORT B
1 054066 016037 000012 001172 MOV RPDS1(RO),$TMP3 ;GET THE DRIVE STATUS REGISTER FROM PORT B
1 054074 013737 001172 001166 MOV $TMP3,$TMP1 ;COPY IT INTO '$TMP1'
1 054102 042737 100100 001166 BIC #ATA!VV,$TMP1 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
1 054110 023737 001164 001166 CMP $TMP0,$TMP1 ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
1 054116 001006 BNE 64$ ;BR IF NOT
1 054120 005737 001164 TST $TMP0 ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
1 054124 001045 BNE 66$ ;BR IF NOT
1 054126 104046 ERROR 46 ;REPORT DRIVE NOT IN NEUTRAL OR NOT SEIZED
1 054130 000137 054330 JMP 68$ ;BYPASS THE REST OF THE CHECKS
1 054134 013737 001170 001126 64$: MOV $TMP2,$BDDAT ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
1 054142 013737 001226 001234 MOV PORTB,PTNBR ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
1 054150 113760 001226 000010 MOVB PORTB,RPCS2(RO) ;SELECT PORT B
1 054156 005737 001164 TST $TMP0 ;SEE IF STATUS EQ 0 FROM PORT A
1 054162 001414 BEQ 65$ ;BR IF ZERO
1 054164 013737 001224 001234 MOV PORTA,PTNBR ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
1 054172 013737 001172 001126 MOV $TMP3,$BDDAT ;'BAD DATA' FOR ERROR TYPE OUT
1 054200 113760 001224 000010 MOVB PORTA,RPCS2(RO) ;SELECT PORT A

```

054206	005737	001166			TST	\$TMP1	:SEE IF STATUS EQ ZERO FROM POR' B.
054212	001012				BNE	66\$:BR IF NOT
054214	012737	177777	001250	65\$:	MOV	#-1,RELEERR	:SET 'RELEASE ERROR' INDICATOR
054222	012760	000011	000000		MOV	#11,RPDS1(RO)	:CLEAR THE DRIVE
054230	012760	000013	000000		MOV	#13,RPDS1(RO)	:RELEASE THE DRIVE
054236	104026				ERROR	26	:TYPE ERROR MESSAGE 26
054240	013737	001170	001126	66\$:	MOV	\$TMP2,\$BDDAT	:LOOK FOR BIT FAILURES WHEN RPDS1 READ
054246	013737	001224	001234		MOV	PORTA,PTNBR	:CHANGE PORT NUMBER
054254	042737	100000	001170		BIC	#ATA,\$TMP2	:DON'T CHECK THE ATTN BIT
054262	023737	001124	001170		CMP	\$GDDAT,\$TMP2	:ALL BITS OK ?
054270	001401				BEQ	67\$:BR IF OK FROM PORT A.
054272	104007				ERROR	7	:REPORT ERROR
054274	013737	001172	001126	67\$:	MOV	\$TMP3,\$BDDAT	:CHECK RPDS1 FOR BIT FAILURES - FROM PORT B.
054302	013737	001226	001234		MOV	PORTB,PTNBR	:CHANGE PORT NUMBER
054310	042737	100000	001172		BIC	#ATA,\$TMP3	:DON'T CHECK THE ATTN BIT
054316	023737	001124	001172		CMP	\$GDDAT,\$TMP3	:SEE IF READ OK FROM POR' B.
054324	001401				BEQ	68\$:BR IF OK
054326	104007				ERROR	7	:REPORT ERROR
054330	000240			68\$:	NOP		
054332	000004				SCOPE		:LOOP ?
054334	000137	056254			JMP	\$EOP	:GO TO END OF TEST

::*****

.SBTTL *** SPECIAL TESTS FOR THE M7775 ('DP') BOARD ***

::*****

::*****

*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.

* 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.

::*****

054340					ST45:	TST	KYBCTL	:PERFORMING ONLY SINGLE TESTS ^
054340	005737	001274				BEQ	2\$:BR IF NOT
054344	001406					BPL	1\$:BR IF JUST ENTERED TEST
054346	100002					JMP	EXEC	:RETURN & GET NEXT TEST NUMBER
054350	000137	002622				JMP	EXEC	:RETURN & GET NEXT TEST NUMBER
054354	012737	177777	001274	1\$:	MOV	#-1,KYBCTL	:SET SINGLE TEST INDICATOR	

93347
93348
93349
93350
93351
93352
93353
93354
93355
93356
93357
93358
93359
93360
93361
93362
93363
93364
93365
93366
93367
93368
93369
93370
93371
93372
93373
93374
93375
93376
93377
93378
93379
93380
93381
93382
93383
93384
93385
93386
93387
93388
93389
93390
93391
93392
93393
93394
93395
93396
93397
93398
93399

```

(3) 054362 112737 000045 001102 25:   MOV      #45,$STNM      ;TEST NUMBER
(3) 054370 012737 054412 001106      MOV      #TEST45,$LPADR ;LOAD LOOP ON TEST ADDRESS
(3) 054376 012737 054412 001110      MOV      #TEST45,$LPERR ;LOAD LOOP ON ERROR ADDRESS
(1) 054404 012737 000004 001176      MOV      #4,$TIMES      ;DO 4 ITERATIONS
9377 054412 012706 001100      TEST45: MOV     #STACK,SP   ;LOAD THE STACK POINTER
9425
(2)                                     ;CLEAR ATTENTION BITS FOR BOTH PORTS
(2)
(2) 054416 113760 001224 000010      MOV      PORTA,RPCS2(RO) ;SELECT PORT #A
(2) 054424 005060 000012 000000      CLR      RPDS1(RO)      ;SEIZE THE DRIVE
(2) 054430 012760 000011 000000      MOV      #11,RPCS1(RO)  ;ISSUE DRIVE CLEAR
(2) 054436 012760 000013 000000      MOV      #13,RPCS1(RO)  ;RELEASE THE DRIVE
(2) 054444 113760 001226 000010      MOV      PORTB,RPCS2(RO) ;SELECT PORT #B
(2) 054452 005060 000012 000000      CLR      RPDS1(RO)      ;SEIZE THE DRIVE THROUGH PORT 'B'
(2) 054456 012760 000011 000000      MOV      #11,RPCS1(RO)  ;ISSUE DRIVE CLEAR
(2) 054464 012760 000013 000000      MOV      #13,RPCS1(RO)  ;RELEASE THE DRIVE
(1)
(2)                                     ;*****
(2)                                     ;SEIZE THE DRIVE THROUGH PORT B
(2)
(2) 054472 113760 001226 000010      MOV      PORTB,RPCS2(RO) ;SELECT PORT B
(2) 054500 013737 001226 001236      MOV      PORTB,SEIZPT   ;STORE SEIZING PORT'S ADDRESS
(2) 054506 005060 000012 000000      CLR      RPDS1(RO)      ;WRITE RPDS1
(2) 054512 013737 001224 001240      MOV      PORTA,OPPR     ;'OPPOSITE' PORT ADDRESS
(2) 054520 113760 001224 000010      MOV      PORTA,RPCS2(RO) ;SELECT PORT A
(2) 054526 013737 001224 001234      MOV      PORTA,PTNBR    ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
(1)
(2)                                     ;*****
(1)                                     ;SET REQUEST THROUGH PORT A
(1)
(1) 054534 005060 000012 000010      CLR      RPDS1(RO)      ;SET REQUEST FOR PORT A
(2) 054540 113760 001226 000010      MOV      PORTB,RPCS2(RO) ;SELECT PORT B
(2) 054546 013737 001226 001234      MOV      PORTB,PTNBR    ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
(1)
(2)                                     ;*****
(2)                                     ;RELEASE THE DRIVE THROUGH PORT B
(2)
(2) 054554 012760 000013 000000      MOV      #13,RPCS1(RO)  ;RELEASE DRIVE THROUGH PORT B
(1)
(2)                                     ;*****
(1)                                     ;WAIT THE MEASURED TIMEOUT FOR THE PORT (+ 25%)
(1)
(1) 054562 013737 001260 001254      MOV      TIMEAP,WATCH   ;SET WATCH TO MEASURED TIMEOUT VALUE + 25%
(2)
(2)                                     ;*****
(2)                                     ;VERIFY THAT THE DRIVE IS STILL SEIZED BY PORT A
(2)
(2) 054570 005037 001244 000000      CLR      CKERR          ;CLEAR THE 'CHECK ERROR' INDICATOR
(2) 054574 016037 000012 001126      MOV      RPDS1(RO), $BDDAT ;GET CONTENTS OF RPDS1
(2) 054602 012737 000012 001122      MOV      #RPDS1,$BDAADR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
(2) 054610 060037 001122 000000      ADD      RO,$BDAADR     ;ADD RH11 BASE ADDRESS
(2) 054614 005037 001124 000000      CLR      $GDDAT        ;WHAT REGISTER SHOULD BE
(2) 054620 023737 001124 001126      CMP      $GDDAT,$BDDAT  ;IS THE REGISTER OK?
(2) 054626 001403 000000 000000      BEQ     66$            ;BR IF OK
(2) 054630 104031 000000 000000      ERROR   31            ;TYPE MESSAGE 31

```



```

(2) 054632 005137 001244          COM      CKERR          ;SET THE REGISTER COMPARE ERPCR INDICATOR
(2) 054636 000240          66$:  NOP
(1) 054640 005737 001244          TST      CKERR          ;REGISTER OK ?
(1) 054644 001402          BEQ      .+6             ;BR IF OK
(1) 054646 000137 055304          JMP      IS              ;BYPASS REST OF TEST IF NOT
(1) 054652 005737 001254          TST      WATCH          ;WATCH EQUAL ZERO ?
(1) 054656 001375          BNE      .-4             ;BR IF NOT

;*****
;CONFIRM THAT THE DRIVE HAS NOT TIMED OUT

(1) 054660 013737 001224 001234    MOV      PORTA,PTNBR     ;PORT NUMBER FOR TYPEOUT
(2) 054666 005037 001244          CLR      CKERR          ;CLEAR THE 'CHECK ERROR' INDICATOR
(2) 054672 016037 000012 001126    MOV      RPDS1(RO),SBDDAT ;GET CONTENTS OF RPDS1
(2) 054700 012737 000012 001122    MOV      #RPDS1,SBDAADR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
(2) 054706 060037 001122          ADD      RO,SBDAADR     ;ADD RH11 BASE ADDRESS
(2) 054712 005037 001124          CLR      $GDDAT         ;WHAT REGISTER SHOULD BE
(2) 054716 023737 001124 001126    CMP      $GDDAT,SBDDAT  ;IS THE REGISTER OK ?
(2) 054724 001403          BEQ      68$           ;BR IF OK
(2) 054726 104035          ERROR   35             ;TYPE MESSAGE 35
(2) 054730 005137 001244          COM      CKERR          ;SET THE REGISTER COMPARE ERPOP INDICATOR
(2) 054734 000240          68$:  NOP
(1) 054736 005737 001244          TST      CKERR          ;REGISTER OK ?
(1) 054742 001402          BEQ      .+6             ;BR IF OK
(1) 054744 000137 055304          JMP      IS              ;BYPASS REST OF TEST IF NOT

;*****
;RELEASE THE DRIVE FROM PORT A

(2) 054750 113760 001224 000010    MOVVB   PORTA,RPCS2(RO) ;SELECT PORT A
(3) 054756 013737 001224 001234    MOV     PORTA,PTNBR     ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
(2) 054764 012760 000013 000000    MOV     #13,RPCS1(RO)  ;ISSUE RELEASE THROUGH PORT A

;VERIFY THAT THE DRIVE IS IN NEUTRAL

(3) 054772 005037 001250          CLR      RELERR        ;CLEAR THE 'RELEASE ERROR' INDICATOR
(3) 054776 012737 000012 001122    MOV      #RPDS1,SBDAOR  ;FORM THE ADDRESS OF RPDS1 FOR TYPEOUT
(3) 055004 060037 001122          ADD      RO,SBDAOR     ;ADD THE I/O BASE ADDRESS
(3) 055010 012737 011700 001124    MOV      #MOL!PGM!DPR!DRY!VV,$GDDAT ;COMPARISON CONSTANT
(3) 055016 113760 001224 000010    MOVVB   PORTA,RPCS2(RO) ;SELECT PORT A.
(3) 055024 016037 000012 001170    MOV      RPDS1(RO),STMP2 ;GET THE DRIVE STATUS REGISTER FROM PORT A.
(3) 055032 013737 001170 001164    MOV      STMP2,STMP0    ;COPY IT INTO 'STMP0'
(3) 055040 042737 100100 001164    BIC     #ATA!VV,STMP0   ;CLEAR PORT DEPENDENT BITS FROM THE COPY
(3) 055046 113760 001226 000010    MOVVB   PORTB,RPCS2(RO) ;SELECT PORT B.
(3) 055054 016037 000012 001172    MOV      RPDS1(RO),STMP3 ;GET THE DRIVE STATUS REGISTER FROM PORT B.
(3) 055062 013737 001172 001166    MOV      STMP3,STMP1    ;COPY IT INTO 'STMP1'
(3) 055070 042737 100100 001166    BIC     #ATA!VV,STMP1   ;CLEAR PORT DEPENDENT BITS FROM THE COPY
(3) 055076 023737 001164 001166    CMP     STMP0,STMP1     ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
(3) 055104 001006          BNE     70$           ;BR IF NOT
(3) 055106 005737 001164          TST     STMP0          ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
(3) 055112 001045          BNE     72$           ;BR IF NOT
(3) 055114 104046          ERROR   46             ;REPORT DRIVE NOT IN NEUTRAL OR NOT SEIZED
(3) 055116 000137 055302          JMP     74$           ;BYPASS THE REST OF THE CHECKS
(3) 055122 013737 001170 001126 70$:  MOV     STMP2,SBDDAT    ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
(3) 055130 013737 001226 001234    MOV     PORTB,PTNBR    ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL

```

```

(3) 055136 113760 001226 000010      MOVB  PORTB,RPCS2(RO)  ;SELECT PORT B.
(3) 055144 005737 001164              TST   $TMP0           ;SEE IF STATUS EQ 0 FROM PORT A.
(3) 055150 001414              BEQ   71$             ;BR IF ZERO
(3) 055152 013737 001224 001234      MOV   PORTA,PTNBR    ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
(3) 055160 013737 001172 001126      MOV   $TMP3,$BDDAT   ;'BAD DATA' FOR ERROR TYPE OUT
(3) 055166 113760 001224 000010      MOVB  PORTA,RPCS2(RO) ;SELECT PORT A.
(3) 055174 005737 001166              TST   $TMP1           ;SEE IF STATUS EQ ZERO FROM PORT B.
(3) 055200 001012              BNE   72$             ;BR IF NOT
(3) 055202 012737 177777 001250 71$:  MOV   #-1,RELERR     ;SET 'RELEASE ERROR' INDICATOR
(3) 055210 012760 000011 000000      MOV   #11,RPCS1(RO)  ;CLEAR THE DRIVE
(3) 055216 012760 000013 000000      MOV   #13,RPCS1(RO)  ;RELEASE THE DRIVE
(3) 055224 104026              ERROR  26            ;TYPE ERROR MESSAGE 26
(3) 055226 013737 001170 001126 72$:  MOV   $TMP2,$BDDAT   ;LOOK FOR BIT FAILURES WHEN RPDS1 PEAD
(3) 055234 013737 001224 001234      MOV   PORTA,PTNBR    ;CHANGE PORT NUMBER
(3) 055242 023737 001124 001170      CMP   $GDDAT,$TMP2   ;ALL BITS OK ?
(3) 055250 001401              BEQ   73$             ;BR IF OK FROM PORT A.
(3) 055252 104007              ERROR  7             ;REPORT ERROR
(3) 055254 013737 001172 001126 73$:  MOV   $TMP3,$BDDAT   ;CHECK RPDS1 FOR BIT FAILURES - FROM PORT B.
(3) 055262 013737 001226 001234      MOV   PORTB,PTNBR    ;CHANGE PORT NUMBER
(3) 055270 023737 001124 001172      CMP   $GDDAT,$TMP3   ;SEE IF READ OK FROM PORT B.
(3) 055276 001401              BEQ   74$             ;BR IF OK
(3) 055300 104007              ERROR  7             ;REPORT ERROR
(3) 055302 000240 74$:  NOP
(1) 055304 000004 1$:  SCOPE ;LOOP ?

```

9446
9447

```

(3) *****
(4) *TEST 46 TEST NO TIMEOUT THROUGH PORT 'B'
(4) *
(4) *VERIFY THAT THE TIMEOUT ONE-SHOT IS NOT TRIGGERED WHEN THE DRIVE
(4) *SWITCHES PORTS AND SEIZING PORT PERFORMS NO REGISTER ACCESSES.
(4) *
(4) * A. SEIZE THE DRIVE THROUGH PORT 'A' BY WRITING 0'S INTO FPDS1.
(4) *
(4) * B. SET PORT REQUEST BY WRITING 0'S INTO RPDS1 FROM PORT 'B'.
(4) *
(4) * C. ISSUE A RELEASE COMMAND FROM PORT 'A'. VERIFY THAT THE DRIVE
(4) *HAS SWITCHED TO THE OTHER PORT AND THAT THE 'ATA' BIT DID NOT
(4) *SET FOR PORT 'A'. REGISTERS WILL NOT BE CHECKED THROUGH PORT 'B'.
(4) *
(4) * D. WAIT THE TIMEOUT INTERVAL + 25%. VERIFY THAT THE DRIVE HAS NOT
(4) *BEEN RELEASED.
(4) *
(4) * E. RELEASE THE DRIVE THROUGH PORT 'B'. VERIFY THAT THE DRIVE
(4) *RETURNED TO NEUTRAL AND THAT NEITHER ATTENTION BIT IS SET.
(4) *
(3) *****

```

```

(2) 055306
(3) 055306 005737 001274      TST   KYBCTL         ;PERFORMING ONLY SINGLE TESTS
(3) 055312 001406              BEQ   2$             ;BR IF NOT
(3) 055314 100002              BPL   1$             ;BR IF JUST ENTERED TEST
(3) 055316 000137 002622      JMP   EXEC           ;RETURN & GET NEXT TEST NUMBER
(3) 055322 012737 177777 001274 1$:  MOV   #-1,KYBCTL     ;SET SINGLE TEST INDICATOR
(3) 055330 112737 000046 001102 2$:  MOVB  #46,$STNM      ;TEST NUMBER
(3) 055336 012737 055360 001106      MOV   #TEST46,$LPADR ;LOAD LOOP ON TEST ADDRESS
(3) 055344 012737 055360 001110      MOV   #TEST46,$LPERR ;LOAD LOOP ON ERROR ADDRESS

```

```

(1) 055352 012737 000004 001176 MOV #4,STIMES ;DO 4 ITERATIONS
9448 055360 012706 001100 TEST46: MOV #STACK,SP ;LOAD THE STACK POINTER
9449
(2) ;CLEAR ATTENTION BITS FOR BOTH PORTS
(2)
(2) 055364 113760 001224 000010 MOVB PORTA,RPCS2(RO) ;SELECT PORT #A
(2) 055372 005060 000012 CLR RPDS1(RO) ;SEIZE THE DRIVE
(2) 055376 012760 000011 000000 MOV #11,RPCS1(RO) ;ISSUE DRIVE CLEAR
(2) 055404 012760 000013 000000 MOV #13,RPCS1(RO) ;RELEASE THE DRIVE
(2) 055412 113760 001226 000010 MOVB PORTB,RPCS2(RO) ;SELECT PORT #B
(2) 055420 005060 000012 CLR RPDS1(RO) ;SEIZE THE DRIVE THROUGH PORT 'B'
(2) 055424 012760 000011 000000 MOV #11,RPCS1(RO) ;ISSUE DRIVE CLEAR
(2) 055432 012760 000013 000000 MOV #13,RPCS1(RO) ;RELEASE THE DRIVE
(1)
(2) ;*****
(2) ;SEIZE THE DRIVE THROUGH PORT A
(2) 055440 113760 001224 000010 MOVB PORTA,RPCS2(RO) ;SELECT PORT A
(2) 055446 013737 001224 001236 MOV PORTA,SEIZPT ;STORE SEIZING PORT'S ADDRESS
(2) 055454 005060 000012 CLR RPDS1(RO) ;WRITE RPDS1
(2) 055460 013737 001226 001240 MOV PORTB,OPPRT ;'OPPOSITE' PORT ADDRESS
(2) 055466 113760 001226 000010 MOVB PORTB,RPCS2(RO) ;SELECT PORT B
(2) 055474 013737 001226 001234 MOV PORTB,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
(1)
(2) ;*****
(2) ;SET REQUEST THROUGH PORT B
(1) 055502 005060 000012 CLR RPDS1(RO) ;SET REQUEST FOR PORT B
(2) 055506 113760 001224 000010 MOVB PORTA,RPCS2(RO) ;SELECT PORT A
(2) 055514 013737 001224 001234 MOV PORTA,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
(1)
(2) ;*****
(1) ;RELEASE THE DRIVE THROUGH PORT A
(1) 055522 012760 000013 000000 MOV #13,RPCS1(RO) ;RELEASE DRIVE THROUGH PORT A
(1)
(2) ;*****
(1) ;WAIT THE MEASURED TIMEOUT FOR THE PORT (+ 25%)
(1) 055530 013737 001266 001254 MOV TIMEBP,WATCH ;SET WATCH TO MEASURED TIMEOUT VALUE + 25%
(2)
(1) ;*****
(1) ;VERIFY THAT THE DRIVE IS STILL SEIZED BY PORT B
(2) 055536 005037 001244 CLR CKERR ;CLEAR THE 'CHECK ERROR' INDICATOR
(2) 055542 016037 000012 001126 MOV RPDS1(RO),SBDDAT ;GET CONTENTS OF RPDS1
(2) 055550 012737 000012 001122 MOV #RPDS1,SBDAOR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
(2) 055556 060037 001122 ADD RO,SBDAOR ;ADD RHI1 BASE ADDRESS
(2) 055562 005037 001124 CLR $GDDAT ;WHAT REGISTER SHOULD BE
(2) 055566 023737 001124 001126 CMP $GDDAT,$BDDAT ;IS THE REGISTER OK ?
(2) 055574 001403 BEQ 66$ ;BR IF OK
(2) 055576 104031 ERROR 31 ;TYPE MESSAGE 31
(2) 055600 005137 001244 COM CKERR ;SET THE REGISTER COMPARE ERROR INDICATOR
(2) 055604 000240 NOP
(1) 055606 005737 001244 TST CKERP ;REGISTER OK ?

```

(1) 055612 001402
(1) 055614 000137 056252
(1) 055620 005737 001254
(1) 055624 001375

BEQ +6 ;BR IF OK
JMP 1\$;BYPASS REST OF TEST IF NOT
TST WATCH ;WATCH EQUAL ZERO ?
BNE -4 ;BR IF NOT

;CONFIRM THAT THE DRIVE HAS NOT TIMED OUT

(1) 055626 013737 001226 001234
(2) 055634 005037 001244
(2) 055640 016037 000012 001126
(2) 055646 012737 000012 001122
(2) 055654 060037 001122
(2) 055660 005037 001124
(2) 055664 023737 001124 001126
(2) 055672 001403
(2) 055674 104035
(2) 055676 005137 001244
(2) 055702 000240
(1) 055704 005737 001244
(1) 055710 001402
(1) 055712 000137 056252

MOV PORTB,PTNBR ;PORT NUMBER FOR TYPEOUT
CLR CKERR ;CLEAR THE 'CHECK ERROR' INDICATOR
MOV RPS1(RO),SBDDAT ;GET CONTENTS OF RPS1
MOV #RPS1,\$B0ADR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
ADD RO,\$B0ADR ;ADD RHL1 BASE ADDRESS
CLR \$GDDAT ;WHAT REGISTER SHOULD BE
CMP \$GDDAT,\$BDDAT ;IS THE REGISTER OK ?
BEQ 68\$;BR IF OK
ERROR 35 ;TYPE MESSAGE 35
COM CKERR ;SET THE REGISTER COMPARE ERROR INDICATOR
68\$: NOP
TST CKERR ;REGISTER OK ?
BEQ +6 ;BR IF OK
JMP 1\$;BYPASS REST OF TEST IF NOT

;RELEASE THE DRIVE FROM PORT B

(3) 055716 113760 001226 000010
(3) 055724 013737 001226 001234
(3) 055732 012760 000013 000000

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

;VERIFY THAT THE DRIVE IS IN NEUTRAL

(3) 055740 005037 001250
(3) 055744 012737 000012 001122
(3) 055752 060037 001122
(3) 055756 012737 011700 001124
(3) 055764 113760 001224 000010
(3) 055772 016037 000012 001170
(3) 056000 013737 001170 001164
(3) 056006 042737 100107 001164
(3) 056014 113760 001226 000010
(3) 056022 016037 000012 001172
(3) 056030 013737 001172 001166
(3) 056036 042737 100100 001166
(3) 056044 023737 001164 001166
(3) 056052 001006
(3) 056054 005737 001164
(3) 056060 001045
(3) 056062 104046
(3) 056064 000137 056250
(3) 056070 013737 001170 001126 70\$:
(3) 056076 013737 001226 001234
(3) 056104 113760 001226 000010
(3) 056112 005737 001164
(3) 056116 001414

CLR RELERR ;CLEAR THE 'RELEASE ERROR' INDICATOR
MOV #RPS1,\$B0ADR ;FORM THE ADDRESS OF RPS1 FOR TYPEOUT
ADD RO,\$B0ADR ;ADD THE I/O BASE ADDRESS
MOV #MOL!PGM!DPR!DRY!VV,\$GDDAT ;COMPARISON CONSTANT
MOVB PORTA,RPCS2(RO) ;SELECT PORT A.
MOV RPS1(RO),STMP2 ;GET THE DRIVE STATUS REGISTER FROM PORT A.
MOV STMP2,STMP0 ;COPY IT INTO 'STMP0'
BIC #ATA!VV,STMP0 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
MOVB PORTB,RPCS2(RO) ;SELECT PORT B.
MOV RPS1(RO),STMP3 ;GET THE DRIVE STATUS REGISTER FROM PORT B.
MOV STMP3,STMP1 ;COPY IT INTO 'STMP1'
BIC #ATA!VV,STMP1 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
CMP STMP0,STMP1 ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
BNE 70\$;BR IF NOT
TST STMP0 ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
BNE 72\$;BR IF NOT
ERROR 46 ;REPORT DRIVE NOT IN NEUTRAL OR NOT SEIZED
JMP 74\$;BYPASS THE REST OF THE CHECKS
MOV STMP2,\$BDDAT ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
MOV PORTB,PTNBR ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
MOVB PORTB,RPCS2(RO) ;SELECT PORT B.
TST STMP0 ;SEE IF STATUS EQ 0 FROM PORT A.
BEQ 71\$;BR IF ZERO

```

(3) 056120 013737 001224 001234 MOV PORTA,PTNBR ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
(3) 056126 013737 001172 001126 MOV $TMP3,$BDDAT ;'BAD DATA' FOR ERROR TYPE OUT
(3) 056134 113760 001224 000010 MOVB PORTA,RPCS2(RO) ;SELECT PORT A.
(3) 056142 005737 001166 TST $TMP1 ;SEE IF STATUS EQ ZERO FROM PORT B.
(3) 056146 001012 BNE 72$ ;BR IF NOT
(3) 056150 012737 177777 001250 71$: MOV 8-1,RELEARR ;SET 'RELEASE ERROR' INDICATOR
(3) 056156 012760 000011 000000 MOV #11,RPCS1(RO) ;CLEAR THE DRIVE
(3) 056164 012760 000013 000000 MOV #13,RPCS1(RO) ;RELEASE THE DRIVE
(3) 056172 104026 ERROR 26 ;TYPE ERROR MESSAGE 26
(3) 056174 013737 001170 001126 72$: MOV $TMP2,$BDDAT ;LOOK FOR BIT FAILURES WHEN RPD51 READ
(3) 056202 013737 001224 001234 MOV PORTA,PTNBR ;CHANGE PORT NUMBER
(3) 056210 023737 001124 001170 CMP $GDDAT,$TMP2 ;ALL BITS OK ?
(3) 056216 001401 BEQ 73$ ;BR IF OK FROM PORT A.
(3) 056220 104007 ERROR 7 ;REPORT ERROR
(3) 056222 013737 001172 001126 73$: MOV $TMP3,$BDDAT ;CHECK RPD51 FOR BIT FAILURES - FROM PORT B.
(3) 056230 013737 001226 001234 MOV PORTB,PTNBR ;CHANGE PORT NUMBER
(3) 056236 023737 001124 001172 CMP $GDDAT,$TMP3 ;SEE IF READ OK FROM PORT B.
(3) 056244 001401 BEQ 74$ ;BR IF OK
(3) 056246 104007 ERROR 7 ;REPORT ERROR
(3) 056250 000240 74$: NOP
(1) 056252 000004 1$: SCOPE ;LOOP ?
9450
9456 .SBTTL END OF PASS ROUTINE
(1)
(2) ;*****
(1) ;*INCREMENT THE PASS NUMBER ($PASS)
(1) ;*INDICATE END-OF-PROGRAM AFTER 1 PASSES THRU THE PROGRAM
(1) ;*TYPE "END PASS #XXXXX TOTAL NUMBER OF ERRORS SINCE LAST REPORT YYYYY"
(1) ;*WHERE XXXXX AND YYYYY ARE DECIMAL NUMBERS
(1) ;*IF THERES A MONITOR GO TO IT
(1) ;*IF THERE ISN'T JUMP TO TST1AA
(1)
(1) SEOP:
(3) 056254 005737 001274 TST KYBCTL ;ENTERED TEST VIA KEYBOARD COMMAND ?
(3) 056260 001402 BEQ +6 ;BR IF NOT
(3) 056262 000137 002622 JMP EXEC ;RETURN TO KEYBOARD CONTROL
(1) 056266 005037 001102 CLR $STNM ;ZERO THE TEST NUMBER
(1) 056272 005037 001176 CLR $TIMES ;ZERO THE NUMBER OF ITERATIONS
(1) 056276 005237 001100 INC $PASS ;INCREMENT THE PASS NUMBER
(1) 056302 042737 100000 001100 BIC #100000,$PASS ;DON'T ALLOW A NEG. NUMBER
(1) 056310 005327 DEC (PC)+ ;LOOP?
(1) 056312 000001 SEOPCT: .WORD 1
(1) 056314 003063 BGT $DOAGN ;YES
(1) 056316 012737 MOV (PC)+,2(PC)+ ;RESTORE COUNTER
(1) 056320 000001 SENDCT: .WORD 1
(1) 056322 056312 SEOPCT
(2) 056324 104401 056332 TYPE 65$ ;TYPE ASCIZ STRING
(2) 056330 000407 BR 64$ ;GET OVER THE ASCIZ
(2) 65$: .ASCIZ <12>'15>'END PASS #/
(2) 64$:
(2) 056350 MOV $PASS,-(SP) ;SAVE $PASS FOR TYPEOUT
(2) 056350 ;TYPE PASS NUMBER
(2) 056354 104405 TYPDS ;GO TYPE--DECIMAL ASCII WITH SIGN
(2) 056356 104401 056364 TYPE 67$ ;TYPE ASCIZ STRING
(2) 056362 000421 BR 66$ ;GET OVER THE ASCIZ

```

```

(2)          ;:67$: .ASCIZ / TOTAL ERRORS SINCE LAST REPORT /
(2) 056426   66$: MOV      SERTTL,-(SP)      ;:SAVE SERTTL FOR TYPEOUT
(2) 056426   013746 001112                ;:TOTAL NUMBER OF ERRORS
(2) 056432   104405                ;:GO TYPE--DECIMAL ASCII WITH SIGN
(1) 056434   104401 001207                ;:TYPE CARRIAGE RETURN, LINE FEED
(1) 056440   005037 001112                ;:CLEAR ERROR TOTAL
(1) 056444   013700 000042                ;:GET MONITOR ADDRESS
(1) 056450   001405                ;:BRANCH IF NO MONITOR
(1) 056452   000005                ;:CLEAR THE WORLD
(1) 056454   004710                ;:GO TO MONITOR
(1) 056456   000240                ;:SAVE ROOM
(1) 056460   000240                ;:FOR
(1) 056462   000240                ;:ACT11
(1) 056464   000137                ;:RETURN
(1) 056466   003114                ;:RETURN
(1) 056470   377 377 000 SRTNAD: .WORD  TST1AA
(1) 056474   056474                ;:NULL CHARACTER STRING

```

```

9457
9458 ;:*****
9459
9460 .SBTTL *** SUBROUTINES ***
9461
9462 ;:*****
9463
9464

```

```

9465 ;ROUTINE TO CHECK FOR KW11-L OR KW11-P CLOCKS
9466 ;IF CLOCK IS PRESENT, THE CLOCK WILL BE STARTED
9467

```

```

9468 056474   012737 056544 000004 CKCLK: MOV      #CKCLK1,#ERRVEC ;SET UP VECTOR FOR CLOCK CHECK
9469 056502   005037 000006          CLR      #ERRVEC+2      ;NEW PSW
9470 056506   005777 122500          TST     #SLKCSR        ;CHECK FOR KW11-P
9471 056512   013701 001216          MOV     $LPVEC,R1      ;KW11-P VECTOR ADDRESS
9472 056516   012721 056626          MOV     #CLOCK,(R1)+  ;SET UP KW11-P VECTOR
9473 056522   012711 000300          MOV     #300,(R1)     ;PSW - PRI 6
9474 056526   012777 177777 122460 MOV     #-1,#SLKCSB    ;LOAD COUNTER BUFFER WITH 1'S
9475 056534   012777 000135 122450 MOV     #135,#SLKCSR   ;SET CLOCK - CNT UP, 16MS. CONT INT
9476 056542   000425                BR      CKCLK3
9477 056544   062706 000004          ADD     #4,SP          ;RESTORE THE STACK POINTER
9478 056550   012737 056606 000004 CKCLK1: MOV     #CKCLK2,#ERRVEC ;CHANGE ERROR VECTOR TO CHECK FOR KW11-L
9479 056556   005777 122436          TST     #SLKS         ;LOOK FOR KW11-L
9480 056562   013701 001222          MOV     $LLVEC,R1     ;KW11-L VECTOR ADDRESS
9481 056566   012721 056626          MOV     #CLOCK,(R1)+  ;SET UP KW11-L VECTOR
9482 056572   012711 000300          MOV     #300,(R1)     ;PSW - PRI 6
9483 056576   012777 000100 122414 MOV     #100,#SLKS    ;SET KW11-L INTERRUPT
9484 056604   000404                BR      CKCLK3
9485 056606   062706 000004          ADD     #4,SP          ;RESTORE THE STACK POINTER
9486 056612   062716 000002          ADD     #2,(SP)       ;INCREMENT RETURN, NO CLOCK
9487 056616   012737 000006 000004 CKCLK3: MOV     #6,#ERRVEC ;RESTORE THE ERROR VECTOR
9488 056624   000207                RTS      PC
9489
9490 ;ROUTINE TO COUNT CLOCK TICKS
9491

```

```

9492 056626   062737 000021 001252 CLOCK: ADD     #17,TIME ;ADD 17 MS TO ELAPSED TIME COUNTER
9493 056634   005737 001254                TST     WATCH         ;IS WATCH ALREADY ZERO

```

```

9494 056640 001406          BEQ     1$          ;BR IF IT IS
9495 056642 162737 000021 001254  SUB     #17.,WATCH ;SUBTRACT 17 MS FROM WATCH DOG COUNTER
9496 056650 100002          BPL     1$          ;BR IF NOT MINUS
9497 056652 005037 001254  CLR     WATCH      ;CLEAR WATCH DOG COUNTER
9498 056656 000002          RTI          ;RETURN

```

9499 ;ROUTINE TO CALCULATE + AND - 25% TIME TOLERANCE VALUES

```

9500
9501
9502 056660 162706 000004  TOLER: SUB     #4,SP ;SETUP STACK
9503 056664 016616 000004  MOV     4(SP), (SP) ;SAVE STACK
9504 056670 013546          MOV     @R5, -(SP) ;GET TIME VALUE
9505 056672 011666 000004  MOV     (SP), 4(SP) ;MOVE TIME VALUE
9506 056676 011666 000006  MOV     (SP), 6(SP) ;MOVE VALUE AGAIN
9507 056702 006216          ASR     (SP)        ;DIVIDE BY 2
9508 056704 006216          ASR     (SP)        ;DIVIDE BY 2 AGAIN (FOR A TOTAL OF 4)
9509 056706 061666 000004  ADD     (SP), 4(SP) ;CALCULATE UPPER LIMIT FOR TIMEOUT
9510 056712 162666 000004  SUB     (SP), 4(SP) ;CALCULATE LOWER LIMIT FOR TIMEOUT
9511 056716 000205          RTS     R5         ;RETURN WITH TOLERANCES ON THE STACK

```

9512 ;*****

9513 .SBTTL 'SYSMAC' UTILITY ROUTINES

9514 ;*****

9515 .SBTTL SCOPE HANDLER ROUTINE

```

9516 ;*****
9517 ;THIS ROUTINE CONTROLS THE LOOPING OF SUBTESTS. IT WILL INCREMENT
9518 ;AND LOAD THE TEST NUMBER($STNM) INTO THE DISPLAY REG.(DISPLAY<7:0)
9519 ;AND LOAD THE ERROR FLAG ($ERFLG) INTO DISPLAY<15:06>
9520 ;THE SWITCH OPTIONS PROVIDED BY THIS ROUTINE ARE:
9521 ;SW14=1      LOOP ON TEST
9522 ;SW11=1      INHIBIT ITERATIONS
9523 ;SW09=1      LOOP ON ERROR
9524 ;CALL
9525 ;*      SCOPE          ;;SCOPE=IOT

```

```

(1) 056720          $SCOPE:
(1) 056720 104407          CKSWR
(1) 056722 032777 040000 122210 1$: BIT     #BIT14, @SWR ;:TEST FOR CHANGE IN SOFT-SWR
(1) 056730 001101          BNE     $OVER      ;:LOOP ON PRESENT TEST?
(1) 056732 000416          ;*****START OF CODE FOR THE XOR TESTER***** ;:YES IF SW14=1
(1) 056734 013746 000004          $XTSTR: BR     6$ ;:IF RUNNING ON THE "XOR" TESTER CHANGE
(1) 056740 012737 056760 000004  MOV     @ERRVEC, -(SP) ;:THIS INSTRUCTION TO A "NOP" (NOP=240)
(1) 056746 005737 177060          MOV     #5$, @ERRVEC ;:SAVE THE CONTENTS OF THE ERROR VECTOR
(1) 056752 012637 000004          TST     @177060 ;:SET FOR TIMEOUT
(1) 056756 000453          MOV     (SP), @ERRVEC ;:TIME OUT ON XOR?
(1) 056760 022626          BR     $$VLAD ;:RESTORE THE ERROR VECTOR
(1) 056762 012637 000004          BR     $$VLAD ;:GO TO THE NEXT TEST
(1) 056766 000413          SS: CMP     (SP), (SP)+ ;:CLEAR THE STACK AFTER A TIME OUT
(1) 056770          MOV     (SP), @ERRVEC ;:RESTORE THE ERROR VECTOR
(1) 056770 105737 001103          BR     7$ ;:LOOP ON THE PRESENT TEST
(1) 056774 001421          6$: ;*****END OF CODE FOR THE XOR TESTER*****
(1) 056774 001421          2$: TSTB   $ERFLG ;:HAS AN ERROR OCCURRED?
(1) 056774 001421          BEQ     3$ ;:BR IF NO

```

```

(1) 056776 123737 001115 001103      CMPB  $ERMAX,$ERFLG  ;; MAX. ERRORS FOR THIS TEST OCCURRED?
(1) 057004 101015      BHI   3$             ;; BR IF NO
(1) 057006 032777 001000 122124      BIT   #BIT09,$SWR    ;; LOOP ON ERROR?
(1) 057014 001404      BEQ   4$             ;; BR IF NO
(1) 057016 013737 001110 001106 7$:  MOV   $LPERR,$LPADR ;; SET LOOP ADDRESS TO LAST SCOPE
(1) 057024 000443      BR    $OVER         ;;
(1) 057026 105037 001103 4$:  CLRB  $ERFLG        ;; ZERO THE ERROR FLAG
(1) 057032 005037 001176      CLR  $TIMES         ;; CLEAR THE NUMBER OF ITERATIONS TO MAKE
(1) 057036 000415      BR    1$            ;; ESCAPE TO THE NEXT TEST
(1) 057040 032777 004000 122072 3$:  BIT   #BIT11,$SWR    ;; INHIBIT ITERATIONS?
(1) 057046 001011      BNE   1$            ;; BR IF YES
(1) 057050 005737 001100      TST  $PASS         ;; IF FIRST PASS OF PROGRAM
(1) 057054 001406      BEQ   1$            ;; INHIBIT ITERATIONS
(1) 057056 005237 001104      INC  $SICNT         ;; INCREMENT ITERATION COUNT
(1) 057062 023737 001176 001104  CMP   $TIMES,$SICNT  ;; CHECK THE NUMBER OF ITERATIONS MADE
(1) 057070 002021      BGE  $OVER         ;; BR IF MORE ITERATION REQUIRED
(1) 057072 012737 000001 001104 1$:  MOV   #1,$SICNT     ;; REINITIALIZE THE ITERATION COUNTER
(1) 057100 013737 057150 001176  MOV   $MXCNT,$TIMES ;; SET NUMBER OF ITERATIONS TO DO
(1) 057106 105237 001102      $SVLAD: INCB  $STNM       ;; COUNT TEST NUMBERS
(1) 057112 011637 001106      MOV  (SP),$LPADR    ;; SAVE SCOPE LOOP ADDRESS
(1) 057116 011637 001110      MOV  (SP),$LPERR    ;; SAVE ERROR LOOP ADDRESS
(1) 057122 005037 001200      CLR  $ESCAPE       ;; CLEAR THE ESCAPE FROM ERROR ADDRESS
(1) 057126 112737 000001 001115  MOVB  #1,$ERMAX     ;; ONLY ALLOW ONE(1) ERROR ON NEXT TEST
(1) 057134 013777 001102 122000 $OVER: MOV  $STNM,$DISPLAY ;; DISPLAY TEST NUMBER
(1) 057142 013716 001106      MOV  $LPADR,(SP)   ;; FUDGE RETURN ADDRESS
(1) 057146 000002      RTI                ;; FIXES PS
(1) 057150 000004      $MXCNT: 4          ;; MAX. NUMBER OF ITERATIONS
9523 .SBTTL ERROR HANDLER ROUTINE

```

```

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

```

```

(1) 057152      $ERROR: CKSWR           ;; TEST FOR CHANGE IN SOFT-SWR
(1) 057152 104407 001102 001242  MOVB  $STNM,$TSTNUM
(2) 057154 113737 001103 7$:  INCB  $ERFLG       ;; SET THE ERROR FLAG
(1) 057162 105237 001103      BEQ   7$           ;; DON'T LET THE FLAG GO TO ZERO
(1) 057166 001775      MOV  $STNM,$DISPLAY ;; DISPLAY TEST NUMBER AND ERROR FLAG
(1) 057170 013777 001102 121744  BIT   #BIT10,$SWR  ;; BELL ON ERROR?
(1) 057176 032777 002000 121734  BEQ   1$           ;; NO - SKIP
(1) 057204 001402      TYPE $SELL        ;; RING BELL
(1) 057206 104401 001202 1$:  INC  $ERTTL       ;; COUNT THE NUMBER OF ERRORS
(1) 057212 005237 001112      MOV  (SP),$ERRPC   ;; GET ADDRESS OF ERROR INSTRUCTION
(1) 057216 011637 001116      SUB  #2,$ERRPC
(1) 057222 162737 000002 001116  MOVB  $ERRPC,$ITEMB ;; STRIP AND SAVE THE ERROR ITEM CODE
(1) 057230 117737 121662 001114  BIT   #BIT13,$SWR  ;; SKIP TYPEOUT IF SET
(1) 057236 032777 020000 121674  BNE  20$          ;; SKIP TYPEOUTS
(1) 057244 001004      JSR  PC,$ERRTYP   ;; GO TO USER ERROR ROUTINE
(1) 057246 004737 057304

```



```

(1) 057252 104401 001207          TYPE      ,SCLF
(1) 057256 005777 121656          20$:      TST      @SWR      ;; HALT ON ERROR
(1) 057262 100002                2$:      BPL      3$          ;; SKIP IF CONTINUE
(1) 057264 000000                ;; HALT ON ERROR!
(1) 057266 104407                ;; TEST FOR CHANGE IN SOFT-SWP
(1) 057270 022737 056454 000042  3$:      CMP      @SENDAD,@#42 ;; ACT-11 AUTO-ACCEPT?
(1) 057276 001001                ;; BRANCH IF NO
(1) 057300 000000                ;; YES
(1) 057302 000002                6$:      RTI          ;; RETURN
(1) 057302 000002                .SBTTL   ERROR MESSAGE TYPEOUT ROUTINE
9524
(1)
(2)
(1)
(1)
(1)
(1)
(1)
(1) 057304 104401 001207          $ERRTYP: TYPE      ,SCLF      ;; "CARRIAGE RETURN" & "LINE FEED"
(1) 057310 010046                MOV      R0,-(SP)    ;; SAVE R0
(1) 057312 005000                CLR      R0          ;; PICKUP THE ITEM INDEX
(1) 057314 153700 001114          BISB    @#$ITEMB,R0
(1) 057320 001004                BNE     1$          ;; IF ITEM NUMBER IS ZERO, JUST
(1)                                MOV      $ERRPC,-(SP) ;; TYPE THE PC OF THE ERROR
(2) 057322 013746 001116                ;; SAVE $ERRPC FOR TYPEOUT
(2)                                ;; ERROR ADDRESS
(2) 057326 104402                TYP0C    ;; GO TYPE--OCTAL ASCII(ALL DIGITS)
(1) 057330 000445                BR      10$         ;; GET OUT
(1) 057332 005300                1$:      DEC      R0          ;; ADJUST THE INDEX SO THAT IT WILL
(1) 057334 006300                ASL     R0          ;; WORK FOR THE ERROR TABLE
(1) 057336 006300                ASL     R0
(1) 057340 006300                ASL     R0
(1) 057342 062700 001304          ADD     @ERRTB,R0   ;; FORM TABLE POINTER
(1) 057346 012037 057356          MOV     (R0)+,2$   ;; PICKUP "ERROR MESSAGE" POINTER
(1) 057352 001404                BEQ     3$          ;; SKIP TYPEOUT IF NO POINTER
(1) 057354 104401                TYPE    ;; TYPE THE "ERROR MESSAGE"
(1) 057356 000000                2$:      .WORD    0      ;; "ERROR MESSAGE" POINTER GOES HERE
(1) 057360 104401 001207          TYPE    ,SCLF      ;; "CARRIAGE RETURN" & "LINE FEED"
(1) 057364 012037 057374          3$:      MOV     (R0)+,4$   ;; PICKUP "DATA HEADER" POINTER
(1) 057370 001404                BEQ     5$          ;; SKIP TYPEOUT IF 0
(1) 057372 104401                TYPE    ;; TYPE THE "DATA HEADER"
(1) 057374 000000                4$:      .WORD    0      ;; "DATA HEADER" POINTER GOES HERE
(1) 057376 104401 001207          TYPE    ,SCLF      ;; "CARRIAGE RETURN" & "LINE FEED"
(1) 057402 010146                5$:      MOV     R1,-(SP)   ;; SAVE R1
(1) 057404 012001                MOV     (R0)+,R1   ;; PICKUP "DATA TABLE" POINTER
(1) 057406 001415                BEQ     9$          ;; BR IF NO DATA TO BE TYPED
(1) 057410 012000                MOV     (R0)+,R0   ;; PICKUP "DATA FORMAT" POINTER
(1) 057412 105720                6$:      TSTB   (R0)+     ;; "OCTAL" OR "DECIMAL"
(1) 057414 001003                BNE     7$          ;; BR IF DECIMAL
(2) 057416 013146                MOV     @R1+,-(SP) ;; SAVE @R1+ FOR TYPEOUT
(2) 057420 104402                TYP0C    ;; GO TYPE--OCTAL ASCII(ALL DIGITS)
(1) 057422 000402                BR      8$
(1) 057424                7$:      MOV     @R1+, SP   ;; SAVE @R1+ FOR TYPEOUT
(2) 057424 013146

```

```

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

```


TYPE ROUTINE

:: BR IF NO--GO POP THE NULL OFF OF STACK
:: GO TYPE A NULL
:: DO NOT COUNT AS A COUNT
:: LOOP

: HORIZONTAL TAB PROCESSOR

:: REPLACE TAB WITH SPACE
:: TYPE A SPACE
:: BRANCH IF NOT AT
:: TAB STOP
:: POP SPACE OFF STACK
:: GET NEXT CHARACTER
:: WAIT UNTIL PRINTER IS READ

:: LOAD CHAR TO BE TYPED ; TO DATA REG.
:: IS CHARACTER A CARRIAGE RETURN?
:: BRANCH IF NO
:: YES--CLEAR CHARACTER COUNT
:: EXIT
:: IS CHARACTER A LINE FEED?
:: BRANCH IF YES
:: COUNT THE CHARACTER
:: CHARACTER COUNT STORAGE

.SBTTL BINARY TO OCTAL (ASCII) AND TYPE

:: *****
:: THIS ROUTINE IS USED TO CHANGE A 16-BIT BINARY NUMBER TO A 6-DIGIT
:: OCTAL (ASCII) NUMBER AND TYPE IT.
:: \$TYPOS---ENTER HERE TO SETUP SUPPRESS ZEROS AND NUMBER OF DIGITS TO TYPE

*CALL:
* MOV NUM,-(SP) :: NUMBER TO BE TYPED
* TYPOS :: CALL FOR TYPEOUT
* .BYTE N :: N=1 TO 6 FOR NUMBER OF DIGITS TO TYPE
* .BYTE M :: M=1 OR 0
* :: 1=TYPE LEADING ZEROS
* :: 0=SUPPRESS LEADING ZEROS

*\$STYPON---ENTER HERE TO TYPE OUT WITH THE SAME PARAMETERS AS THE LAST
*\$TYPOS OR \$TYPOC

*CALL:
* MOV NUM,-(SP) :: NUMBER TO BE TYPED
* STYPON :: CALL FOR TYPEOUT

*\$TYPOC---ENTER HERE FOR TYPEOUT OF A 16 BIT NUMBER

*CALL:
* MOV NUM,-(SP) :: NUMBER TO BE TYPED
* TYPOC :: CALL FOR TYPEOUT

\$TYPOS: MOV 2(SP),-(SP) :: PICKUP THE MODE
MOV 1(SP),\$OFILL :: LOAD ZERO FILL SWITCH
MOVB (SP)+,\$OMODE+1 :: NUMBER OF DIGITS TO TYPE
ADD #2,(SP) :: ADJUST RETURN ADDRESS
BR \$STYPON

(1) 057570 002770
(1) 057572 004737 057630
(1) 057576 105337 057674
(1) 057602 000770

057604 112716 000040 8\$: MOV 1(SP),-(SP)
057610 004737 057630 9\$: JSR PC,\$TYPEC
057614 132737 000007 057674 BITB #7,\$CHARCNT
057622 001372 BNE 9\$
057624 005726 TST (SP)+
057626 000724 BR 2\$
057630 105777 121314 \$TYPEC: TSTB 2\$TPS
057634 100375 BPL \$TYPEC
057636 116677 000002 121306 MOV 2(SP),2\$TPB
057644 122766 000015 000002 CMPB #CR,2(SP)
057652 001003 BNE 1\$
057654 105037 057674 CLRB \$CHARCNT
057660 000406 BR \$TYPEX
057662 122766 000012 000002 1\$: CMPB #LF,2(SP)
057670 001402 BEQ \$TYPEX
057672 105227 INCB (PC)+
057674 000000 \$CHARCNT: .WORD 0
057676 000207 \$TYPEX: RTS PC

9526

```

(1) 057724 112737 000001 060123 $TYPOC: MOVB #1,$OFILL           ;; SET THE ZERO FILL SWITCH
(1) 057732 112737 000006 060125           MOVB #6,$OMODE+1       ;; SET FOR SIX(6) DIGITS
(1) 057740 112737 000005 060122 $TYPON: MOVB #5,$OCNT     ;; SET THE ITERATION COUNT
(1) 057746 010346           MOV R3,-(SP)         ;; SAVE R3
(1) 057750 010446           MOV R4,-(SP)         ;; SAVE R4
(1) 057752 010546           MOV R5,-(SP)         ;; SAVE R5
(1) 057754 113704 060125           MOVB $OMODE+1,R4     ;; GET THE NUMBER OF DIGITS TO TYPE
(1) 057760 005404           NEG R4
(1) 057762 062704 000006           ADD #6,R4           ;; SUBTRACT IT FOR MAX. ALLOWED
(1) 057766 110437 060124           MOVB R4,$OMODE       ;; SAVE IT FOR USE
(1) 057772 113704 060123           MOVB $OFILL,R4       ;; GET THE ZERO FILL SWITCH
(1) 057776 016605 000012           MOV 12(SP),R5        ;; PICKUP THE INPUT NUMBER
(1) 060002 005003           CLR R3              ;; CLEAR THE OUTPUT WORD
(1) 060004 006105           1$: ROL R5           ;; ROTATE MSB INTO "C"
(1) 060006 000404           BR 3$              ;; GO DO MSB
(1) 060010 006105           2$: ROL R5           ;; FORM THIS DIGIT
(1) 060012 006105           ROL R5
(1) 060014 006105           ROL R5
(1) 060016 010503           MOV R5,R3
(1) 060020 006103           3$: ROL R3           ;; GET LSB OF THIS DIGIT
(1) 060022 105337 060124           DECB $OMODE          ;; TYPE THIS DIGIT?
(1) 060026 100016           BPL 7$             ;; BR IF NO
(1) 060030 042703 177770           BIC #177770,R3      ;; GET RID OF JUNK
(1) 060034 001002           BNE 4$             ;; TEST FOR 0
(1) 060036 005704           TST R4             ;; SUPPRESS THIS 0?
(1) 060040 001403           BEQ 5$             ;; BR IF YES
(1) 060042 005204           4$: INC R4          ;; DON'T SUPPRESS ANYMORE 0'S
(1) 060044 052703 000060           BIS #'0,R3          ;; MAKE THIS DIGIT ASCII
(1) 060050 052703 000040           5$: BIS #' ,R3      ;; MAKE ASCII IF NOT ALREADY
(1) 060054 110337 060120           MOVB R3,$S          ;; SAVE FOR TYPING
(1) 060060 104401 060120           TYPE $S            ;; GO TYPE THIS DIGIT
(1) 060064 105337 060122           7$: DECB $OCNT      ;; COUNT BY 1
(1) 060070 003347           BGT 2$            ;; BR IF MORE TO DO
(1) 060072 002402           BLT 6$            ;; BR IF DONE
(1) 060074 005204           INC R4             ;; INSURE LAST DIGIT ISN'T A BLANK
(1) 060076 000744           BR 2$             ;; GO DO THE LAST DIGIT
(1) 060100 012605           6$: MOV (SP)+,R5     ;; RESTORE R5
(1) 060102 012604           MOV (SP)+,R4       ;; RESTORE R4
(1) 060104 012603           MOV (SP)+,R3       ;; RESTORE R3
(1) 060106 016666 000002 000004           MOV 2(SP),4(SP)    ;; SET THE STACK FOR RETURNING
(1) 060114 012616           MOV (SP)+,(SP)
(1) 060116 000002           RTI               ;; RETURN
(1) 060120 000           8$: .BYTE 0         ;; STORAGE FOR ASCII DIGIT
(1) 060121 000           .BYTE 0           ;; TERMINATOR FOR TYPE ROUTINE
(1) 060122 000           $OCNT: .BYTE 0     ;; OCTAL DIGIT COUNTER
(1) 060123 000           $OFILL: .BYTE 0    ;; ZERO FILL SWITCH
(1) 060124 000000           $OMODE: .WORD 0    ;; NUMBER OF DIGITS TO TYPE
9527 .SBTTL CONVERT BINARY TO DECIMAL AND TYPE ROUTINE

```

1
2
1
1
1
1
1
1

```

*****
*THIS ROUTINE IS USED TO CHANGE A 16-BIT BINARY NUMBER TO A 5-DIGIT
*SIGNED DECIMAL (ASCII) NUMBER AND TYPE IT. DEPENDING ON WHETHER THE
*NUMBER IS POSITIVE OR NEGATIVE A SPACE OR A MINUS SIGN WILL BE TYPED
*BEFORE THE FIRST DIGIT OF THE NUMBER. LEADING ZEROS WILL ALWAYS BE
*REPLACED WITH SPACES.
*CALL:

```

```

(1)          :+      MOV      NUM,-(SP)      ::PUT THE BINARY NUMBER ON THE STACK
(1)          :+      TYPDS      ::GO TO THE ROUTINE
(1)          STIPDS:
(1) 060126   010046   MOV      R0,-(SP)      ::PUSH R0 ON STACK
(3) 060126   010046   MOV      R1,-(SP)      ::PUSH R1 ON STACK
(3) 060130   010146   MOV      R2,-(SP)      ::PUSH R2 ON STACK
(3) 060132   010246   MOV      R3,-(SP)      ::PUSH R3 ON STACK
(3) 060134   010346   MOV      R5,-(SP)      ::PUSH R5 ON STACK
(3) 060136   010546   MOV      #20200,-(SP)  ::SET BLANK SWITCH AND SIGN
(1) 060140   012746   MOV      20(SP),R5    ::GET THE INPUT NUMBER
(1) 060144   016605   BPL      1$          ::BR IF INPUT IS POS.
(1) 060150   100004   NEG      R5          ::MAKE THE BINARY NUMBER POS.
(1) 060152   005405   MOVVB   #'-,1(SP)   ::MAKE THE ASCII NUMBER NEG.
(1) 060154   112766   CLR      R0          ::ZERO THE CONSTANTS INDEX
(1) 060162   005000   MOV      #DBLK,R3    ::SETUP THE OUTPUT POINTER
(1) 060164   012703   MOVVB   #' ,(R3)+   ::SET THE FIRST CHARACTER TO A BLANK
(1) 060170   112723   CLR      R2          ::CLEAR THE BCD NUMBER
(1) 060174   005002   MOV      $DTBL(R0),R1::GET THE CONSTANT
(1) 060176   016001   SUB      R1,R5      ::FORM THIS BCD DIGIT
(1) 060202   160105   BLT     4$          ::BR IF DONE
(1) 060204   002402   INC     R2          ::INCREASE THE BCD DIGIT BY 1
(1) 060206   005202   BR      3$
(1) 060210   000774   ADD     R1,R5      ::ADD BACK THE CONSTANT
(1) 060212   060105   TST     R2          ::CHECK IF BCD DIGIT=0
(1) 060214   005702   BNE     5$          ::FALL THROUGH IF 0
(1) 060216   001002   TSTB   (SP)        ::STILL DOING LEADING 0'S?
(1) 060220   105716   BMI     7$          ::BR IF YES
(1) 060222   100407   ASLB   (SP)        ::MSD?
(1) 060224   106316   BCC     6$          ::BR IF NO
(1) 060226   103003   MOVVB   1(SP),-1(R3) ::YES--SET THE SIGN
(1) 060230   116663   BIS     #'0,R2      ::MAKE THE BCD DIGIT ASCII
(1) 060236   052702   BIS     #' ,R2      ::MAKE IT A SPACE IF NOT ALREADY A DIGIT
(1) 060242   052702   MOVVB   R2,(R3)+   ::PUT THIS CHARACTER IN THE OUTPUT BUFFER
(1) 060244   110223   TST     (R0)+      ::JUST INCREMENTING
(1) 060250   005720   CMP     R0,#10     ::CHECK THE TABLE INDEX
(1) 060252   020027   BLT     2$          ::GO DO THE NEXT DIGIT
(1) 060256   002746   BGT     8$          ::GO TO EXIT
(1) 060260   003002   MOV     R5,R2      ::GET THE LSD
(1) 060262   010502   BR      6$          ::GO CHANGE TO ASCII
(1) 060264   000764   TSTB   (SP)+      ::WAS THE LSD THE FIRST NON-ZERO?
(1) 060266   105726   BPL     9$          ::BR IF NO
(1) 060270   100003   MOVVB   -1(SP),-2(R3)::YES--SET THE SIGN FOR TYPING
(1) 060272   116663   CLRB   (R3)        ::SET THE TERMINATOR
(3) 060300   105013   MOV     (SP)+,R5   ::POP STACK INTO R5
(3) 060302   012605   MOV     (SP)+,R3   ::POP STACK INTO R3
(3) 060304   012603   MOV     (SP)+,R2   ::POP STACK INTO R2
(3) 060306   012602   MOV     (SP)+,R1   ::POP STACK INTO R1
(3) 060310   012601   MOV     (SP)+,R0   ::POP STACK INTO R0
(3) 060312   012600   TYPE   $DBLK      ::NOW TYPE THE NUMBER
(1) 060314   104401   MOV     2(SP),4(SP)::ADJUST THE STACK
(1) 060320   016666   MOV     (SP)+,(SP)
(1) 060326   012616   RTI
(1) 060330   000002   ::RETURN TO USER
(1) 060332   023420   $DTBL: 10000.
(1) 060334   001750   1000.
(1) 060336   000144   100.

```

(1) 060340 000012
(1) 060342 000004
9528
(1)
(2)
(1)
(1) 060352 000000
(1) 060354 000000
(1) 060356 000000
(1) 060360 000001
(1) 060361 060361
(1) 060362 060362

10.
\$DBLK: .BLKW 4
.SBTTL TTY INPUT ROUTINE

.ENABL LSB
\$TKCNT: .WORD 0 ;: NUMBER OF ITEMS IN QUEUE
\$TKQIN: .WORD 0 ;: INPUT POINTER
\$TKQOUT: .WORD 0 ;: OUTPUT POINTER
\$TKQSRT: .BLKB 1 ;: TTY KEYBOARD QUEUE
\$TKQEND=.
.EVEN

;*TK INITIALIZE ROUTINE
;*THIS ROUTINE WILL INITIALIZE THE TTY KEYBOARD INPUT QUEUE
;*SETUP THE INTERRUPT VECTOR AND TURN ON THE KEYBOARD INTERRUPT

;*CALL:
* JSR PC,\$TKINT
* RETURN

(1) 060362 005037 060352
(1) 060366 012737 060360 060354
(1) 060374 013737 060354 060356
(1) 060402 012737 060432 000060
(1) 060410 012737 000200 000062
(1) 060416 005777 120524
(1) 060422 012777 000100 120514
(1) 060430 000207

\$TKINT: CLR \$TKCNT ;: CLEAR COUNT OF ITEMS IN QUEUE
MOV \$TKQSRT,\$TKQIN ;: MOVE THE STARTING ADDRESS OF THE
MOV \$TKQIN,\$TKQOUT ;: QUEUE INTO THE INPUT & OUTPUT POINTERS.
MOV \$TKSRV,\$TKVEC ;: INITIALIZE THE KEYBOARD VECTOR
MOV #200,\$TKVEC+2 ;: "BR" LEVEL 4
TST \$TKB ;: CLEAR DONE FLAG
MOV #100,\$TKS ;: ENABLE TTY KEYBOARD INTERRUPT
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.

(1) 060432 117746 120510
(1) 060436 042716 177600
(1) 060442 021627 000007
(1) 060446 001004
(1) 060450 022737 000176 001140
(1) 060456 001500

\$TKSRV: MOVB \$TKB, -(SP) ;: PICKUP THE CHARACTER
BIC #177,(SP) ;: STRIP THE JUNK
1\$: CMP (SP),#7 ;: IS IT A CONTROL G?
BNE 2\$;: BRANCH IF NO
CMP #SWREG,SWR ;: IS SOFT-SWR SELECTED?
BEQ 6\$;: GO TO SWR CHANGE

(1) 060460
(1) 060460 022737 000001 060352
(1) 060466 001004
(1) 060470 104401 001202
(1) 060474 005726
(1) 060476 000451
(1) 060500 021627 000023
(1) 060504 001021
(1) 060506 005077 120432
(1) 060512 005726
(1) 060514 105777 120424
(1) 060520 100375
(1) 060522 117746 120420
(1) 060526 042716 177600

2\$: CMP #1,\$TKCNT ;: IS THE QUEUE FULL?
BNE 3\$;: BRANCH IF NO
TYPE \$BELL ;: RING THE TTY BELL
TST (SP)+ ;: CLEAN CHARACTER OFF OF STACK
BR 5\$;: EXIT
3\$: CMP (SP),#23 ;: IS IT A CONTROL-S?
BNE 32\$;: BRANCH IF NO
CLR \$TKS ;: DISABLE TTY KEYBOARD INTERRUPTS
TST (SP)+ ;: CLEAN CHAR OFF STACK
31\$: TSTB \$TKS ;: WAIT FOR A CHAR
BPL 31\$;: LOOP UNTIL ITS THERE
MOVB \$TKB, -(SP) ;: GET THE CHARACTER
BIC #177,(SP) ;: MAKE IT 7-BIT ASCII

```

(1) 060532 022627 000021      CMP      (SP)+,#21      ;; IS IT A CONTROL-G?
(1) 060536 001366          BNE      31$           ;; BRANCH IF NO
(1) 060540 012777 000100 120376  MOV      #100,#$TKS    ;; REENABLE TTY KEYBOARD INTERRUPTS
(1) 060546 000002          RTI                    ;; RETURN
(1) 060550 005237 060352 32$: INC      $TKCNT        ;; COUNT THIS CHARACTER
(1) 060554 021627 000140      CMP      (SP),#140     ;; IS IT UPPER CASE?
(1) 060560 002405          BLT      4$           ;; BRANCH IF YES
(1) 060562 021627 000175      CMP      (SP),#175     ;; IS IT A SPECIAL CHAR?
(1) 060566 003002          BGT      4$           ;; BRANCH IF YES
(1) 060570 042716 000040      BIC      #40,(SP)      ;; MAKE IT UPPER CASE
(1) 060574 112677 177554 4$: MOVB   (SP)+,$STKQIN  ;; AND PUT IT IN QUEUE
(1) 060600 005237 060354      INC      $TKQIN        ;; UPDATE THE POINTER
(1) 060604 023727 060354 060361  CMP      $TKQIN,#$TKQEND ;; GO OFF THE END?
(1) 060612 001003          BNE      5$           ;; BRANCH IF NO
(1) 060614 012737 060360 060354  MOV      #$TKQ5RT,$STKQIN ;; RESET THE POINTER
(1) 060622 000002 5$: RTI                    ;; RETURN
    
```

```

*****
; SOFTWARE SWITCH REGISTER CHANGE ROUTINE.
; ROUTINE IS ENTERED FROM THE TRAP HANDLER, AND WILL
; SERVICE THE TEST FOR CHANGE IN SOFTWARE SWITCH REGISTER TRAP
; CALL WHEN OPERATING IN TTY INTERRUPT MODE.
    
```

```

(1) 060624 022737 000176 001140 $CKSWR: CMP      #SWREG,SWR    ;; IS THE SOFT-SWR SELECTED
(1) 060632 001104          BNE      15$          ;; EXIT IF NOT
(1) 060634 105777 120304      TSTB    $TKS          ;; IS A CHAR WAITING?
(1) 060640 100101          BPL      15$          ;; IF NOT, EXIT
(1) 060642 117746 120300      MOVB    $TKB,-(SP)    ;; YES
(1) 060646 042716 177600      BIC     #177,(SP)    ;; MAKE IT 7-BIT ASCII
(1) 060652 021627 000007      CMP     (SP),#7      ;; IS IT A CONTROL-G?
(1) 060656 001300          BNE      2$           ;; IF NOT, PUT IT IN THE TTY QUEUE
(1)                                ;; AND EXIT
    
```

```

*****
; CONTROL IS PASSED TO THIS POINT FROM EITHER THE TTY INTERRUPT SERVICE
; ROUTINE OR FROM THE SOFTWARE SWITCH REGISTER TRAP CALL, AS A RESULT OF A
; CONTROL-G BEING TYPED, AND THE SOFTWARE SWITCH REGISTER BEING SELECTED.
    
```

```

(1) 060660 123727 001134 000001 6$: CMPB   $AUTOB,#1    ;; ARE WE RUNNING IN AUTO-MODE?
(1) 060666 001674          BEQ     2$           ;; BRANCH IF YES
(1) 060670 005726          TST    (SP)+         ;; CLEAR CONTROL-G OFF STACK
(1) 060672 004737 060362      JSR    PC,$TKINT     ;; FLUSH THE TTY INPUT QUEUE
(1) 060676 005077 120242      CLR    $TKS          ;; DISABLE TTY KEYBOARD INTERRUPTS
(1) 060702 112737 000001 001135  MOVB   #1,$INTAG     ;; SET INTERRUPT MODE INDICATOR
(1) 060710 104401 061466      TYPE   ,$CNTLG       ;; ECHO THE CONTROL-G (1G)
(1) 060714 104401 061473      TYPE   ,$MSWR        ;; TYPE CURRENT CONTENTS
(2) 060720 013746 000176      MOV    $WREG,-(SP)   ;; SAVE SWREG FOR TYPEOUT
(2) 060724 104402          TYPOC             ;; GO TYPE--OCTAL ASCII(ALL DIGITS)
(1) 060726 104401 061504      TYPE   ,$MNEW        ;; PROMPT FOR NEW SWR
(1) 060732 005046 19$: CLR    -(SP)        ;; CLEAR COUNTER
(1) 060734 005046          CLR    -(SP)        ;; THE NEW SWR
(1) 060736 105777 120202 7$: TSTB   $TKS          ;; CHAR THERE?
(1) 060742 100375          BPL    7$           ;; IF NOT TRY AGAIN
(1) 060744 117746 120176      MOVB   $TKB,-(SP)   ;; PICK UP CHAR
(1) 060750 042716 177600      BIC    #177,(SP)    ;; MAKE IT 7-BIT ASCII
    
```

```

(1)
(1) 060754 021627 000025 95: CMP (SP),#25 ;: IS IT A CONTROL-U?
(1) 060760 001005 BNE 105 ;: BRANCH IF NOT
(1) 060762 104401 061461 TYPE ,SCNTLU ;: YES, ECHO CONTROL-U 'U'
(1) 060766 062706 000006 205: ADD #6 SP ;: IGNORE PREVIOUS INPUT
(1) 060772 000757 BR 195 ;: LET S TRY IT AGAIN

(1)
(1) 060774 021627 000015 105: CMP (SP),#15 ;: IS IT A <CR>?
(1) 061000 001022 BNE 165 ;: BRANCH IF NO
(1) 061002 005766 000004 TST 4(SP) ;: YES, IS IT THE FIRST CHAR?
(1) 061006 001403 BEQ 115 ;: BRANCH IF YES
(1) 061010 016677 000002 120122 MOV 2(SP),2SWR ;: SAVE NEW SWR
(1) 061016 062706 000006 115: ADD #6 SP ;: CLEAR UP STACK
(1) 061022 104401 001207 145: TYPE ,SCRLF ;: ECHO <CR> AND <LF>
(1) 061026 123727 001135 000001 CMPB $INTAG,#1 ;: RE-ENABLE TTY KBD INTERRUPTS?
(1) 061034 001003 BNE 155 ;: BRANCH IF NOT
(1) 061036 012777 000100 120100 MOV #100,2STKS ;: RE-ENABLE TTY KBD INTERRUPTS
(1) 061044 000002 155: RTI ;: RETURN
(1) 061046 004737 057630 165: JSR PC,$TYPEC ;: ECHO CHAR
(1) 061052 021627 000060 CMP (SP),#60 ;: CHAR < 0?
(1) 061056 002420 BLT 185 ;: BRANCH IF YES
(1) 061060 021627 000067 CMP (SP),#67 ;: CHAR > 7?
(1) 061064 003015 BGT 185 ;: BRANCH IF YES
(1) 061066 042726 000060 BIC #60,(SP)+ ;: STRIP-OFF ASCII
(1) 061072 005766 000002 TST 2(SP) ;: IS THIS THE FIRST CHAR
(1) 061076 001403 BEQ 175 ;: BRANCH IF YES
(1) 061100 006316 ASL (SP) ;: NO, SHIFT PRESENT
(1) 061102 006316 ASL (SP) ;: CHAR OVER TO MAKE
(1) 061104 006316 ASL (SP) ;: ROOM FOR NEW ONE.
(1) 061106 005266 000002 175: INC 2(SP) ;: KEEP COUNT OF CHAR
(1) 061112 056616 177776 BIS -2(SP),(SP) ;: SET IN NEW CHAR
(1) 061116 000707 BR 75 ;: GET THE NEXT ONE
(1) 061120 104401 001206 185: TYPE ,SQUES ;: TYPE ?<CR><LF>
(1) 061124 000720 BR 205 ;: SIMULATE CONTROL-U
.DSABL LSB

```

```

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

```

```

(1) 061126 011646 $RDCHR: MOV (SP),-(SP) ;: PUSH DOWN THE PC AND
(1) 061130 016666 000004 000002 MOV 4(SP),2(SP) ;: THE PS
(1) 061136 005066 000004 CLR 4(SP) ;: GET READY FOR A CHARACTER
(2) 061142 005046 CLR -(SP) ;: PUT NEW PS ON STACK
(2) 061144 012746 061152 MOV #64$,-(SP) ;: PUT NEW PC ON STACK
(2) 061150 000002 RTI ;: POP NEW PC AND PS
(1) 061152 005737 060352 645: TST $TKCNT ;: WAIT ON A CHARACTER
(1) 061156 001775 BEQ 15

```



```

(1) 061160 005337 060352          DEC      STKCNT          ;; DECREMENT THE COUNTER
(1) 061164 117766 177166 000004    MOVB     #STKQOUT,4(SP) ;; GET ONE CHARACTER
(1) 061172 005237 060356 060361    INC      STKQOUT        ;; UPDATE THE POINTER
(1) 061176 023727 060356          CMPB     STKQOUT,#STKQEND ;; DID IT GO OFF OF THE END?
(1) 061204 001003                    BNE     2$             ;; BRANCH IF NO
(1) 061206 012737 060360 060356    MOV      #STKQSR,STKQOUT ;; RESET THE POINTER
(1) 061214 000002                    RTI                      ;; RETURN
(2)                                     ;; *****
(1)                                     ;; THIS ROUTINE WILL INPUT A STRING FROM THE TTY
(1)                                     ;; CALL:
(1)                                     ;; *
(1)                                     ;; *   RDLIN          ;; INPUT A STPING FROM THE TTY
(1)                                     ;; *   RETURN HERE  ;; ADDRESS OF FIRST CHARACTER WILL BE ON THE STACK
(1)                                     ;; *                                     ;; TERMINATOR WILL BE A BYTE OF ALL 0'S
(1) 061216 010346          $RDLIN: MOV      R3,-(SP)          ;; SAVE R3
(1) 061220 005046          CLR      -(SP)          ;; CLEAR THE RUBOUT KEY
(1) 061222 012703 061452          1$: MOV      #STTYIN,R3      ;; GET ADDRESS
(1) 061226 022703 061461          2$: CMPB     #STTYIN+7,R3    ;; BUFFER FULL?
(1) 061232 101456          BLOS    4$             ;; BR IF YES
(1) 061234 104410          RDCHR   (SP)+,(R3)      ;; GO READ ONE CHARACTER FROM THE TTY
(1) 061236 112613          MOVB    (SP)+,(R3)      ;; GET CHARACTER
(1) 061240 122713 000177          10$: CMPB    #177,(R3)     ;; IS IT A RUBOUT
(1) 061244 001022          BNE     5$             ;; BR IF NO
(1) 061246 005716          TST     (SP)           ;; IS THIS THE FIRST RUBOUT?
(1) 061250 001007          BNE     6$             ;; BR IF NO
(1) 061252 112737 000134 061450          MOVB    #' \ ,9$        ;; TYPE A BACK SLASH
(1) 061260 104401 061450          TYPE    ,9$
(1) 061264 012716 177777          MOV     #-1,(SP)       ;; SET THE RUBOUT KEY
(1) 061270 005303          6$: DEC      R3          ;; BACKUP BY ONE
(1) 061272 020327 061452          CMP     R3,#STTYIN     ;; STACK EMPTY?
(1) 061276 103434          BLOS    4$             ;; BR IF YES
(1) 061300 111337 061450          MOVB    (R3),9$        ;; SETUP TO TYPEOUT THE DELETED CHAR.
(1) 061304 104401 061450          TYPE    ,9$
(1) 061310 000746          BR      2$             ;; GO TYPE
(1) 061312 005716          5$: TST     (SP)           ;; RUBOUT KEY SET?
(1) 061314 001406          BEQ     7$             ;; BR IF NO
(1) 061316 112737 000134 061450          MOVB    #' \ ,9$        ;; TYPE A BACK SLASH
(1) 061324 104401 061450          TYPE    ,9$
(1) 061330 005016          CLR     (SP)          ;; CLEAR THE RUBOUT KEY
(1) 061332 122713 000025          7$: CMPB    #25,(R3)     ;; IS CHARACTER A CTRL U?
(1) 061336 001003          BNE     8$             ;; BR IF NO
(1) 061340 104401 061461          TYPE    ,SCNTLU        ;; TYPE A CONTROL "U"
(1) 061344 000726          BR      1$             ;; GO START OVER
(1) 061346 122713 000022          8$: CMPB    #22,(R3)     ;; IS CHARACTER A "+R"?
(1) 061352 001011          BNE     3$             ;; BRANCH IF NO
(1) 061354 105013          CLRB   (R3)          ;; CLEAR THE CHARACTER
(1) 061356 104401 001207          TYPE    ,SCRLF        ;; TYPE A "CR" & "LF"
(1) 061362 104401 061452          TYPE    ,STTYIN       ;; TYPE THE INPUT STRING
(1) 061366 000717          BR      2$             ;; GO PICKUP ANOTHER CHACTER
(1) 061370 104401 001206          4$: TYPE    ,SQUES        ;; TYPE A '?'
(1) 061374 000712          BR      1$             ;; CLEAR THE BUFFER AND LOOP
(1) 061376 111337 061450          3$: MOVB    (R3),9$        ;; ECHO THE CHARACTER
(1) 061402 104401 061450          TYPE    ,9$
(1) 061406 122723 000015          CMPB    #15,(R3)+     ;; CHECK FOR RETURN
(1) 061412 001305          BNE     2$             ;; LOOP IF NOT RETURN
(1) 061414 105063 177777          CLRB   -1(R3)        ;; CLEAR RETURN (THE 15)

```

```

(1) 061420 104401 001210          TYPE      $LF           ;; TYPE A LINE FEED
(1) 061424 005726          TST      (SP)+        ;; CLEAN RUBOUT KEY FROM THE STACK
(1) 061426 012603          MOV      (SP)+,R3     ;; RESTORE R3
(1) 061430 011646          MOV      (SP),-(SP)   ;; ADJUST THE STACK AND PUT ADDRESS OF THE
(1) 061432 016666 000004 000002  MOV      4(SP),2(SP)   ;; FIRST ASCII CHARACTER ON IT
(1) 061440 012766 061452 000004  MOV      @STTYIN,4(SP)
(1) 061446 000002          RTI                    ;; RETURN
(1) 061450 000          9$: .BYTE      0           ;; STORAGE FOR ASCII CHAR. TO TYPE
(1) 061451 000          .BYTE      0           ;; TERMINATOR
(1) 061452 000007          $TTYIN: .BLKB      7       ;; RESERVE 7 BYTES FOR TTY INPUT
(1) 061461 136 006525 000012  $CNTLU: .ASCIZ  /?U/<15><12> ;; CONTROL "U"
(1) 061466 043536 005015 000  $CNTLG: .ASCIZ  /?G/<15><12> ;; CONTROL "G"
(1) 061473 015 051412 051127  $MSWR:  .ASCIZ  <15><12>/SWR = /
(1) 061500 036440 000040          $MNEW:  .ASCIZ  / NEW = /
(1) 061504 020040 042516 020127
(1) 061512 020075 000
(1) 061516 0361516
9529
(1) .EVEN
(1) .SBTTL READ AN OCTAL NUMBER FROM THE TTY
(1)
(1) *****
(1) *THIS ROUTINE WILL READ AN OCTAL (ASCII) NUMBER FROM THE TTY AND
(1) *CHANGE IT TO BINARY.
(1) *THE INPUT CHARACTERS WILL BE CHECKED TO INSURED THEY ARE LEGAL
(1) *OCTAL DIGITS. IF AN ILLEGAL CHARACTER IS READ A "?" WILL BE TYPED
(1) *FOLLOWED BY A CARRIAGE RETURN-LINE FEED. THE COMPLETE NUMBER MUST
(1) *THEN BE RETYPED. THE INPUT IS TERMINATED BY TYPING A CARRIAGE RETURN.
(1) *CALL:
(1) *
(1) *      RDOCT
(1) *      RETURN HERE
(1) *
(1) *      ;; READ AN OCTAL NUMBER
(1) *      ;; LOW ORDER BITS ARE ON TOP OF THE STACK
(1) *      ;; HIGH ORDER BITS ARE IN $HIOCT
(1)
(1) 061516 011646          $RDOCT: MOV      (SP),-(SP)   ;; PROVIDE SPACE FOR THE
(1) 061520 016666 000004 000002  MOV      4(SP),2(SP)   ;; INPUT NUMBER
(3) 061526 010046          MOV      R0,-(SP)     ;; PUSH R0 ON STACK
(3) 061530 010146          MOV      R1,-(SP)     ;; PUSH R1 ON STACK
(3) 061532 010246          MOV      R2,-(SP)     ;; PUSH R2 ON STACK
(1) 061534 104411          1$: RDLIN           ;; READ AN ASCII LINE
(1) 061536 012600          MOV      (SP)+,R0     ;; GET ADDRESS OF 1ST CHARACTER
(1) 061540 010037 061644          MOV      R0,$$        ;; AND SAVE IT
(1) 061544 005001          CLR      R1           ;; CLEAR DATA WORD
(1) 061546 005002          CLR      R2
(1) 061550 112046          2$: MOV8      (R0)+,-(SP)   ;; PICKUP THIS CHARACTER
(1) 061552 001420          BEQ      3$           ;; IF ZERO GET OUT
(1) 061554 122716 000060          CMPB    #'0,(SP)     ;; MAKE SURE THIS CHARACTER
(1) 061560 003026          BGT      4$           ;; IS AN OCTAL DIGIT
(1) 061562 122716 000067          CMPB    #'7,(SP)
(1) 061566 002423          BLT      4$
(1) 061570 006301          ASL      R1           ;; *2
(1) 061572 006102          ROL      R2           ;; *4
(1) 061574 006301          ASL      R1           ;; *4
(1) 061576 006102          ROL      R2           ;; *8
(1) 061600 006301          ASL      R1           ;; *8
(1) 061602 006102          ROL      R2
(1) 061604 042716 177770          BIC      #'C7,(SP)   ;; STRIP THE ASCII JUNK
(1) 061610 062601          ADD     (SP)+,R1     ;; ADD IN THIS DIGIT
(1) 061612 000756          BR      2$           ;; LOOP
    
```

(1) 061614 005726
 (1) 061616 010166 000012
 (1) 061622 010237 061654
 (3) 061626 012602
 (3) 061630 012601
 (3) 061632 012600
 (1) 061634 000002
 (1) 061636 005726
 (1) 061640 105010
 (1) 061642 104401
 (1) 061644 000000
 (1) 061646 104401 001206
 (1) 061652 000730
 (1) 061654 000000

9530

```

3$:  TST      (SP)+      ;; CLEAN TERMINATOR FROM STACK
      MOV     R1,12(SP)  ;; SAVE THE RESULT
      MOV     R2,$SHIOCT
      MOV     (SP)+,R2   ;; POP STACK INTO R2
      MOV     (SP)+,R1   ;; POP STACK INTO R1
      MOV     (SP)+,R0   ;; POP STACK INTO R0
      RTI
4$:  TST      (SP)+      ;; CLEAN PARTIAL FROM STACK
      CLRB   (R0)        ;; SET A TERMINATOR
      TYPE   ;; TYPE UP THRU THE BAD CHAR.
5$:  .WORD   0
      TYPE   $QUES      ;; "?" "CR" & "LF"
      BR     1$         ;; TRY AGAIN
$SHIOCT: .WORD 0        ;; HIGH ORDER BITS GO HERF
.SBTTL  SAVE AND RESTORE R0-R5 ROUTINES

```

```

*****
*SAVE R0-R5
*CALL:
*   SAVREG
*UPON RETURN FROM $SAVREG THE STACK WILL LOOK LIKE:
*
*TOP---(+16)
* +2---(+18)
* +4---R5
* +6---R4
* +8---R3
*+10---R2
*+12---R1
*+14---R0

```

(1) 061656
 (3) 061656 010046
 (3) 061660 010146
 (3) 061662 010246
 (3) 061664 010346
 (3) 061666 010446
 (3) 061670 010546
 (1) 061672 016646 000022
 (1) 061676 016646 000022
 (1) 061702 016646 000022
 (1) 061706 016646 000022
 (1) 061712 000002

```

$SAVREG:
      MOV     R0,-(SP)   ;; PUSH R0 ON STACK
      MOV     R1,-(SP)   ;; PUSH R1 ON STACK
      MOV     R2,-(SP)   ;; PUSH R2 ON STACK
      MOV     R3,-(SP)   ;; PUSH R3 ON STACK
      MOV     R4,-(SP)   ;; PUSH R4 ON STACK
      MOV     R5,-(SP)   ;; PUSH R5 ON STACK
      MOV     22(SP),-(SP) ;; SAVE PS OF MAIN FLOW
      MOV     22(SP),-(SP) ;; SAVE PC OF MAIN FLOW
      MOV     22(SP),-(SP) ;; SAVE PS OF CALL
      MOV     22(SP),-(SP) ;; SAVE PC OF CALL
      RTI

```

(1) 061714
 (1) 061714 012666 000022
 (1) 061720 012666 000022
 (1) 061724 012666 000022
 (1) 061730 012666 000022
 (3) 061734 012605
 (3) 061736 012604
 (3) 061740 012603
 (3) 061742 012602

```

*RESTORE R0-R5
*CALL:
*   RESREG
$RESREG:
      MOV     (SP)+,22(SP) ;; RESTORE PC OF CALL
      MOV     (SP)+,22(SP) ;; RESTORE PS OF CALL
      MOV     (SP)+,22(SP) ;; RESTORE PC OF MAIN FLOW
      MOV     (SP)+,22(SP) ;; RESTORE PS OF MAIN FLOW
      MOV     (SP)+,R5     ;; POP STACK INTO R5
      MOV     (SP)+,R4     ;; POP STACK INTO R4
      MOV     (SP)+,R3     ;; POP STACK INTO R3
      MOV     (SP)+,R2     ;; POP STACK INTO R2

```

```

(3) 061744 012601      MOV      (SP)+,R1      ;;POP STACK INTO R1
(3) 061746 012600      MOV      (SP)+,R0      ;;POP STACK INTO R0
(1) 061750 000002      RTI
9531 .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.

```

```

(1) 061752 010046      $TRAP: MOV      RO, -(SP)      ;;SAVE RO
(1) 061754 016600 000002 MOV      2(SP),R0      ;;GET TRAP ADDRESS
(1) 061760 005740      TST      -(R0)        ;;BACKUP BY 2
(1) 061762 111000      MOVB     (R0),R0      ;;GET RIGHT BYTE OF TRAP
(1) 061764 006300      ASL     R0            ;;POSITION FOR INDEXING
(1) 061766 016000 062006 MOV      $TRPAD(R0),R0 ;;INDEX TO TABLE
(1) 061772 000200      RTS      R0           ;;GO TO ROUTINE

```

```

;;THIS IS USE TO HANDLE THE "GETPRI" MACRO

```

```

(1) 061774 011646      $TRAP2: MOV     (SP), -(SP)    ;;MOVE THE PC DOWN
(1) 061776 016666 000004 000002 MOV     4(SP),2(SP)    ;;MOVE THE PSW DOWN
(1) 062004 000002      RTI      ;;RESTORE THE PSW

```

```

.SBTTL TRAP TABLE

```

```

;THIS TABLE CONTAINS THE STARTING ADDRESSES OF THE ROUTINES CALLED
;BY THE "TRAP" INSTRUCTION.

```

```

; ROUTINE
;-----
(3) 062006 061774      $TRPAD: .WORD  $TRAP2      TRAP+1(104401) TTY TYPEOUT ROUTINE
(3) 062010 057460      $TYPE  ;;CALL=TYPE      TRAP+2(104402) TYPE OCTAL NUMBER (WITH LEADING ZEROS)
(3) 062012 057724      $TYPOC ;;CALL=TYPOC     TRAP+3(104403) TYPE OCTAL NUMBER (NO LEADING ZEROS)
(3) 062014 057700      $TYPOS ;;CALL=TYPOS     TRAP+4(104404) TYPE OCTAL NUMBER (AS PER LAST CALL)
(3) 062016 057740      $TYPON ;;CALL=TYPON     TRAP+5(104405) TYPE DECIMAL NUMBER (WITH SIGN)
(3) 062020 060126      $TYPDS ;;CALL=TYPDS
(3) 062022 060714      $GTSWR ;;CALL=GTSWR      TRAP+6(104406) GET SOFT-SWR SETTING
(3) 062024 060624      $CKSWR ;;CALL=CKSWR      TRAP+7(104407) TEST FOR CHANGE IN SOFT-SWR
(3) 062026 061126      $RDCHR ;;CALL=RDCHR      TRAP+10(104410) TTY TYPEIN CHARACTER ROUTINE
(3) 062030 061216      $RDLIN ;;CALL=RDLIN      TRAP+11(104411) TTY TYPEIN STRING ROUTINE
(3) 062032 061516      $RDOCT ;;CALL=RDOCT      TRAP+12(104412) READ AN OCTAL NUMBER FROM TTY
(3) 062034 061656      $SAVREG ;;CALL=SAVREG     TRAP+13(104413) SAVE RO-RS ROUTINE
(3) 062036 061714      $RESREG ;;CALL=RESREG     TRAP+14(104414) RESTORE RO-RS ROUTINE

```

```

;*****

```

```

.SBTTL TELETYPE MESSAGES

```

```

;*****

```

```

9539 062040 005015 055012 026532 TITLE: .ASCII CR LF LF ZZ-CZRJE-C CR LF

```

	062046	055103	045122	026505	
	062054	006503	012		
9540	062057	122	030120	027464	.ASCIZ SRP04'S 6 DUAL CONTROLLER LOGIC TEST - PART 10<CR><LF><LF>
	062064	027465	020066	052504	
	062072	046101	041440	047117	
	062100	051124	046117	042514	
	062106	020122	047514	044507	
	062114	020103	042524	052123	
	062122	026440	050040	051101	
	062130	020124	006461	005012	
	062136	000			
9541	062137	015	042412	052116	ENTERA: .ASCIZ <CR><LF>/ENTER DRIVE ADDRESS: /
	062144	051105	042040	044522	
	062152	042526	040440	042104	
	062160	042522	051523	020072	
	062166	000			
9542	062167	111	053116	046101	ADRERR: .ASCIZ /INVALID ADDRESS/<CR><LF>
	062174	042111	040440	042104	
	062202	042522	051523	005015	
	062210	000			
9543	062211	015	050012	051117	PORTAIS: .ASCIZ <CR><LF>/PORT A ADDRESS IS: /
	062216	020124	020101	042101	
	062224	051104	051505	020123	
	062232	051511	020072	000	
9544	062237	015	050012	051117	PORTBIS: .ASCIZ <CR><LF>/PORT B ADDRESS IS: /
	062244	020124	020102	042101	
	062252	051104	051505	020123	
	062260	051511	020072	000	
9545	062265	015	051412	051531	NOCLOCK: .ASCIZ <CR><LF>/SYSTEM MUST HAVE 'L' OR 'P' CLOCK/<CR><LF><LF>
	062272	042524	020115	052515	
	062300	052123	044040	053101	
	062306	020105	046047	020047	
	062314	051117	023440	023520	
	062322	041440	047514	045503	
	062330	005015	000012		
9546	062334	042412	052116	051105	TESTNO: .ASCIZ <LF>/ENTER TEST #: /
	062342	052040	051505	020124	
	062350	035043	000040		
9547	062354	047111	040526	044514	BADNO: .ASCIZ /INVALID TEST NUMBER/<CR><LF>
	062362	020104	042524	052123	
	062370	047040	046525	042502	
	062376	006522	000012		
9548	062402	042440	051122	051117	TSTERR: .ASCIZ / ERRORS/<CR><LF>
	062410	006523	000012		
9549	062414	005015	052012	042510	ADDRIS: .ASCIZ <CR><LF><LF>/THE PRESENT ADDRESS OF THE RH11 (RPCS1) IS:
	062422	050040	042522	042523	
	062430	052116	040440	042104	
	062436	042522	051523	047440	
	062444	020106	044124	020105	
	062452	044122	030461	024040	
	062460	050122	051503	024461	
	062466	044440	035123	000040	
9550	062474	042412	052116	051105	NTPH11: .ASCIZ <LF>/ENTER NEW RH11 ADDRESS: .
	062502	047040	053505	051040	
	062510	030510	020061	042101	
	062516	051104	051505	035123	

```

062524 000040
9551
9552
9553
9554
9555
9556
9557
9558 062526 051127 047117 020107 EM1: .ASCIZ /WRONG DRIVE TYPE/
      062534 051104 053111 020105
      062542 054524 042520 000
9559
9560 062547 104 044522 042526 EM2: .ASCIZ /DRIVE NOT ON LINE/
      062554 047040 052117 047440
      062562 020116 044514 042516
      062570 000
9561
9562 062571 123 051105 040511 EM3: .ASCIZ /SERIAL NUMBER READ THROUGH EACH PORT NOT THE SAME/
      062576 020114 052516 041115
      062604 051105 051040 040505
      062612 020104 044124 047522
      062620 043525 020110 040505
      062626 044103 050040 051117
      062634 020124 047516 020124
      062642 044124 020105 040523
      062650 042515 000
9563
9564 062653 104 044522 042526 EM4: .ASCIZ /DRIVE NOT SEIZED BY PORT/
      062660 047040 052117 051440
      062666 044505 042532 020104
      062674 054502 050040 051117
      062702 000124
9565
9566 062704 051127 047117 020107 EM5: .ASCIZ /WRONG STATUS SEEN BY THE SEIZING PORT/
      062712 052123 052101 051525
      062720 051440 042505 020116
      062726 054502 052040 042510
      062734 051440 044505 044532
      062742 043516 050040 051117
      062750 000124
9567
9568 062752 042522 044507 052123 EM6: .ASCIZ /REGISTER CONTENTS WERE SEEN BY OPPOSITE PORT - DRIVE WAS SEIZED
      062760 051105 041440 047117
      062766 042524 052116 020123
      062774 042527 042522 051440
      063002 042505 020116 054502
      063010 047440 050120 051517
      063016 052111 020105 047520
      063024 052122 026440 042040
      063032 044522 042526 053440
      063040 051501 051440 044505
      063046 042532 000104
9569
9570 063052 042522 044507 052123 EM7: .ASCIZ /REGISTER CONTENTS WRONG AFTER RELEASE OR TIMEOUT
      063060 051105 041440 047117
      063066 042524 052116 020123

```

	063074	051127	047117	020107		
	063102	043101	042524	020122		
	063110	042522	042514	051501		
	063116	020105	051117	052040		
	063124	046511	047505	052125		
	063132	000				
9571						
9572	063133	122	043505	051511	EM10:	.ASCIZ REGISTER CONTENTS WRONG
	063140	042524	020122	047503		
	063146	052116	047105	051524		
	063154	053440	047522	043516		
	063162	000				
9573						
9574	063163	103	047117	051124	EM11:	.ASCIZ CONTROL BUS PARITY ERROR READING INDICATED REGISTER
	063170	046117	041040	051525		
	063176	050040	051101	052111		
	063204	020131	051105	047522		
	063212	020122	042522	042101		
	063220	047111	020107	047111		
	063226	044504	040503	042524		
	063234	020104	042522	044507		
	063242	052123	051105	000		
9575						
9576	063247	104	044522	042526	EM12:	.ASCIZ DRIVE NOT SEIZED BY DRIVE CLEAR COMMAND
	063254	047040	052117	051440		
	063262	044505	042532	020104		
	063270	054502	042040	044522		
	063276	042526	041440	042514		
	063304	051101	041440	046517		
	063312	040515	042116	000		
9577						
9578	063317	122	040505	044504	EM13:	.ASCIZ /READIN PRESET DOES NOT SET VOLUME VALID FOR THE PORT
	063324	020116	051120	051505		
	063332	052105	042040	042517		
	063340	020123	047516	020124		
	063346	042523	020124	047526		
	063354	052514	042515	053040		
	063362	046101	042111	043040		
	063370	051117	052040	042510		
	063376	050040	051117	000124		
9579						
9580	063404	047526	052514	042515	EM14:	.ASCIZ /VOLUME VALID SET ON THE WRONG PORT
	063412	053040	046101	042111		
	063420	051440	052105	047440		
	063426	020116	044124	020105		
	063434	051127	047117	020107		
	063442	047520	052122	000		
9581						
9582	063447	101	052124	020116	EM15:	.ASCIZ ATTN BIT WRONG AFTER TIMEOUT - REQUEST NOT SET
	063454	044502	020124	051127		
	063462	047117	020107	043101		
	063470	042524	020122	044524		
	063476	042515	052517	020124		
	063504	020055	042522	052521		
	063512	051505	020124	047516		
	063520	020124	042523	000124		

3583	063526	052101	047124	041040	EM16:	.ASCIZ	.ATTN BIT WRONG AFTER RELEASE - REQUEST SET
3584	063534	052111	053440	047522			
	063542	043516	040440	052106			
	063550	051105	051040	046105			
	063556	040505	042523	026440			
	063564	051040	050505	042525			
	063572	052123	051440	052105			
	063600	000					
3585	063601	101	052124	020116	EM17:	.ASCIZ	.ATTN BIT WRONG AFTER RELEASE - REQUEST NOT SET
3586	063606	044502	020124	051127			
	063614	047117	020107	043101			
	063622	042524	020122	042522			
	063630	042514	051501	020105			
	063636	020055	042522	052521			
	063644	051505	020124	047516			
	063652	020124	042523	000124			
3587	063660	051104	053111	020105	EM20:	.ASCIZ	DRIVE NOT SEIZED WHEN ATTN BIT FOR PORT CLEARED
3588	063666	047516	020124	042523			
	063674	055111	042105	053440			
	063702	042510	020116	052101			
	063710	047124	041040	052111			
	063716	043040	051117	050040			
	063724	051117	020124	046103			
	063732	040505	042522	000104			
3589	063740	051104	053111	020105	EM21:	.ASCIZ	DRIVE SEIZED WHEN ZERO WRITTEN IN ATTN BIT
3590	063746	042523	055111	042105			
	063754	053440	042510	020116			
	063762	042532	047522	053440			
	063770	044522	052124	047105			
	063776	044440	020116	052101			
	064004	047124	041040	052111			
	064012	000					
3591	064013	104	044522	042526	EM22:	.ASCIZ	DRIVE NOT IN NEUTRAL AFTER TIMEOUT - REQUEST NOT SET
3592	064020	047040	052117	044440			
	064026	020116	042516	052125			
	064034	040522	020114	043101			
	064042	042524	020122	044524			
	064050	042515	052517	020124			
	064056	020055	042522	052521			
	064064	051505	020124	047516			
	064072	020124	042523	000124			
3593	064100	044524	042515	052517	EM23:	.ASCIZ	TIMEOUT CLEARED THE DRIVE'S ERROR BIT
3594	064106	020124	046103	040505			
	064114	042522	020104	044124			
	064122	020105	051104	053111			
	064130	023505	020123	051105			
	064136	047522	020122	044502			
	064144	000124					
3595	064146	042522	042514	051501	EM24:	.ASCIZ	RELEASE COMMAND RELEASED DRIVE WITH ERRORS SET

	064154	020105	047503	046515		
	064162	047101	020104	042522		
	064170	042514	051501	042105		
	064176	042040	044522	042526		
	064204	053440	052111	020110		
	064212	051105	047522	051522		
	064220	051440	052105	000		
9597						
9598	064225	124	046511	047505	EM25:	.ASCIZ TIMEOUT ONE-SHOT DID NOT RETRIGGER
	064232	052125	047440	042516		
	064240	051455	047510	020124		
	064246	044504	020104	047516		
	064254	020124	042522	051124		
	064262	043511	042507	000122		
9599						
3600	064270	051104	053111	020105	EM26:	.ASCIZ DRIVE NOT IN NEUTRAL AFTER RELEASE - REQUEST NOT SET
	064276	047516	020124	047111		
	064304	047040	052505	051124		
	064312	046101	040440	052106		
	064320	051105	051040	046105		
	064326	040505	042523	026440		
	064334	051040	050505	042525		
	064342	052123	047040	052117		
	064350	051440	052105	000		
9601						
9602	064355	122	043505	051511	EM27:	.ASCIZ REGISTER WRONG AFTER RELEASE WITH REQUEST SET
	064362	042524	020122	051127		
	064370	047117	020107	043101		
	064376	042524	020122	042522		
	064404	042514	051501	020105		
	064412	044527	044124	051040		
	064420	050505	042525	052123		
	064426	051440	052105	000		
9603						
9604	064433	104	044522	042526	EM30:	.ASCIZ DRIVE SEIZED BY RELEASE COMMAND ISSUED WHEN DRIVE IN NEUTRAL
	064440	051440	044505	042532		
	064446	020104	054502	051040		
	064454	046105	040505	042523		
	064462	041440	046517	040515		
	064470	042116	044440	051523		
	064476	042525	020104	044127		
	064504	047105	042040	044522		
	064512	042526	044440	020116		
	064520	042516	052125	040522		
	064526	000114				
9605						
9606	064530	051104	053111	020105	EM31:	.ASCIZ DRIVE IN NEUTRAL AFTER RELEASE - REQUEST SET
	064536	047111	047040	052505		
	064544	051124	046101	040440		
	064552	052106	051105	051040		
	064560	046105	040505	042523		
	064566	026440	051040	050505		
	064574	042525	052123	051440		
	064602	052105	000			
9607						
9608	064605	101	052124	020116	EM32:	.ASCIZ ATTN BIT WRONG AFTER RECALIBRATE COMMAND

064612	044502	020124	051127
064620	047117	020107	043101
064626	042524	020122	042522
064634	040503	044514	051102
064642	052101	020105	047503
064650	046515	047101	000104

9609
9610

064656	051104	053111	020105
064664	042522	052524	047122
064672	042105	052040	020117
064700	042516	052125	040522
064706	020114	043111	042040
064714	044522	042526	041440
064722	042514	051101	043440
064730	053111	047105	053440
064736	044510	042514	042040
064744	044522	042526	051440
064752	044505	042532	000104

EM33: .ASCIZ DRIVE RETURNED TO NEUTRAL IF DRIVE CLEAR GIVEN WHILE DRIVE SEIZED/

9611
9612

064760	051104	053111	020105
064766	042522	052524	047122
064774	042105	052040	020117
065002	042516	052125	040522
065010	020114	043111	046440
065016	051501	041123	051525
065024	044440	044516	020124
065032	044507	042526	020116
065040	044127	046111	020105
065046	051104	053111	020105
065054	042523	055111	042105
065062	000		

EM34: .ASCIZ DRIVE RETURNED TO NEUTRAL IF MASSBUS INIT GIVEN WHILE DRIVE SEIZED/

9613

9615	065063	124	046511	047505	EM35: .ASCIZ /TIMEOUT ONE SHOT FIRED WITHOUT REGISTER ACCESS
	065070	052125	047440	042516	
	065076	051440	047510	020124	
	065104	044506	042522	020104	
	065112	044527	044124	052517	
	065120	020124	042522	044507	
	065126	052123	051105	040440	
	065134	041503	051505	000123	
9616	065142	044524	042515	052517	EM36: .ASCIZ /TIMEOUT HAS NOT OCCURRED WITHIN 2 SECONDS
9617	065150	020124	040510	020123	
	065156	047516	020124	041517	
	065164	052503	051122	042105	
	065172	053440	052111	044510	
	065200	020116	020062	042523	
	065206	047503	042116	000123	
9618	065214	051104	053111	020105	EM37: .ASCIZ /DRIVE IS NON-EXISTENT ('NED' BIT SET)
9619	065222	051511	047040	047117	
	065230	042455	044530	052123	
	065236	047105	020124	023450	
	065244	042516	023504	041040	
	065252	052111	051440	052105	
	065260	000051			
9620	065262	052101	047124	041040	EM40: .ASCIZ /ATTN BIT FOR PORT NO1 RESET BY MASSBUS CLEAR
9621	065270	052111	043040	051117	
	065276	050040	051117	020124	
	065304	047516	020124	042522	
	065312	042523	020124	054502	
	065320	046440	051501	041123	
	065326	051525	041440	042514	
	065334	051101	000		
9622	065337	124	046511	047505	EM41: .ASCIZ /TIMEOUT CLEARED THE ATTENTION BIT
9623	065344	052125	041440	042514	
	065352	051101	042105	052040	
	065360	042510	040440	052124	
	065366	047105	044524	047117	
	065374	041040	052111	000	
9624	065401	104	044522	042526	EM42: .ASCIZ /DRIVE NOT IN NEUTRAL OR SEIZED AFTER ATTN BIT WRITTEN
9625	065406	047040	052117	044440	
	065414	020116	042516	052125	
	065422	040522	020114	051117	
	065430	051440	044505	042532	
	065436	020104	043101	042524	
	065444	020122	052101	047124	
	065452	041040	052111	053440	
	065460	044522	052124	047105	
	065466	000			
9626	065467	104	044522	042526	EM43: .ASCIZ /DRIVE IN NEUTRAL AFTER ATTENTION BIT WRITTEN
9627	065474	044440	020116	042516	
	065502	052125	040522	020114	
	065510	043101	042524	020122	

	065516	052101	042524	052116	
	065524	047511	020116	044502	
	065532	020124	051127	052111	
	065540	042524	000116		
9628					
9629	065544	051127	052111	020105	EM44: .ASCIZ WRITE ATTENTION BIT DID NOT SET PORT REQUEST
	065552	052101	042524	052116	
	065560	047511	020116	044502	
	065566	020124	044504	020104	
	065574	047516	020124	042523	
	065602	020124	047520	052122	
	065610	051040	050505	042525	
	065616	052123	000		
9630					
9631	065621	103	047117	051124	EM45: .ASCIZ @CONTROLLER SELECT SWITCH ON DRIVE NOT IN 'A-B'
	065626	046117	042514	020122	
	065634	042523	042514	052103	
	065642	051440	044527	041524	
	065650	020110	047117	042040	
	065656	044522	042526	047040	
	065664	052117	044440	020116	
	065672	040447	041057	000047	
9632					
9633	065700	040503	023516	020124	EM46: .ASCIZ /CAN'T ACCESS DRIVE THROUGH EITHER PORT/
	065706	041501	042503	051523	
	065714	042040	044522	042526	
	065722	052040	051110	052517	
	065730	044107	042440	052111	
	065736	042510	020122	047520	
	065744	052122	000		
9634					
9635	065747	101	052124	020116	EM47: .ASCIZ /ATTN BIT FOR SEIZING PORT NOT CLEARED BY MASSBUS INIT/
	065754	044502	020124	047506	
	065762	020122	042523	055111	
	065770	047111	020107	047520	
	065776	052122	047040	052117	
	066004	041440	042514	051101	
	066012	042105	041040	020131	
	066020	040515	051523	052502	
	066026	020123	047111	052111	
	066034	000			
9636					
9637	066035	101	052124	020116	EM50: .ASCIZ /ATTN BIT FOR OPPOSITE PORT CLEARED BY MASSBUS INIT/
	066042	044502	020124	047506	
	066050	020122	050117	047520	
	066056	044523	042524	050040	
	066064	051117	020124	046103	
	066072	040505	042522	020104	
	066100	054502	046440	051501	
	066106	041123	051525	044440	
	066114	044516	000124		
9638					
9639	066120	052101	047124	041040	EM51: .ASCIZ /ATTN BIT CLEARED BY MASSBUS INIT, DRIVE IN NEUTRAL/
	066126	052111	041440	042514	
	066134	051101	042105	041040	
	066142	020131	040515	051523	

	066150	052502	020123	047111	
	066156	052111	020054	051104	
	066164	053111	020105	047111	
	066172	047040	052505	051124	
	066200	046101	000		
9640					
9641	066203	124	042510	040440	EMS2: .ASCIZ THE ATTN BIT IS SET AFTER TIMEOUT WITH NO REQUEST & 'EPP' SET
	066210	052124	020116	044502	
	066216	020124	051511	051440	
	066224	052105	040440	052106	
	066232	051105	052040	046511	
	066240	047505	052125	053440	
	066246	052111	020110	047516	
	066254	051040	050505	042525	
	066262	052123	023040	023440	
	066270	051105	023522	051440	
	066276	052105	000		
9642					
9643	066301	103	047101	052047	EMS3: .ASCIZ /CAN'T READ THE ATTN BIT FROM THE 'OPPOSITE' PORT/
	066306	051040	040505	020104	
	066314	044124	020105	052101	
	066322	047124	041040	052111	
	066330	043040	047522	020115	
	066336	044124	020105	047447	
	066344	050120	051517	052111	
	066352	023505	050040	051117	
	066360	000124			
9644					
9645	066362	042522	042514	051501	EMS4: .ASCIZ /RELEASE COMMAND RECOGNIZED WHEN ISSUED BY NON-SEIZING PORT
	066370	020105	047503	046515	
	066376	047101	020104	042522	
	066404	047503	047107	055111	
	066412	042105	053440	042510	
	066420	020116	051511	052523	
	066426	042105	041040	020131	
	066434	047516	026516	042523	
	066442	055111	047111	020107	
	066450	047520	052122	000	
9646					
9647	066455	124	046511	047505	EMS5: .ASCIZ /TIMEOUT ONE-SHOT IS LESS THAN 500 MS.
	066462	052125	047440	042516	
	066470	051455	047510	020124	
	066476	051511	046040	051505	
	066504	020123	044124	047101	
	066512	032440	030060	046440	
	066520	000123			
9648					
9649	066522	044122	030461	042040	EMS6: .ASCIZ /RH11 DIDN'T RESPOND TO ADDRESSING/
	066530	042111	023516	020124	
	066536	042522	050123	047117	
	066544	020104	047524	040440	
	066552	042104	042522	051523	
	066560	047111	000107		
9650					
9651					
9652					

Line	Reg	PC	PC	PC	Msg	Port	Port	Reg	Reg
9653									
9654									
9655	066564	042524	052123	021440	DH1: .ASCIZ /TEST # ERR PC PORT # REG ADR CONTENTS/				
	066572	020040	051105	020122					
	066600	041520	020040	047520					
	066606	052122	021440	020040					
	066614	042522	020107	042101					
	066622	020122	047503	052116					
	066630	047105	051524	000					
9656	066635	124	051505	020124	DH3: .ASCIZ /TEST # ERR PC REG ADR PORT A PORT B/				
	066642	020043	042440	051122					
	066650	050040	020103	051040					
	066656	043505	040440	051104					
	066664	050040	051117	020124					
	066672	020101	050040	051117					
	066700	020124	000102						
9657	066704	020040	020040	020040	DH4: .ASCII / SEIZE ERROR <CR> <LF>				
	066712	020040	020040	020040					
	066720	020040	020040	042523					
	066726	055111	020105	020040					
	066734	051105	047522	006522					
	066742	012							
9658	066743	124	051505	020124	.ASCIZ /TEST # ERR PC PORT # PORT # REG ADR GOOD BAD/				
	066750	020043	042440	051122					
	066756	050040	020103	050040					
	066764	051117	020124	020043					
	066772	050040	051117	020124					
	067000	020043	051040	043505					
	067006	040440	051104	043440					
	067014	047517	020104	020040					
	067022	041040	042101	000					
9659	067027	124	051505	020124	DH5: .ASCIZ /TEST # ERR PC PORT # REG ADR GOOD BAD/				
	067034	020043	042440	051122					
	067042	050040	020103	050040					
	067050	051117	020124	020043					
	067056	051040	043505	040440					
	067064	051104	043440	047517					
	067072	020104	020040	041040					
	067100	042101	000						
9660	067103	040	020040	020040	DH7: .ASCII / RELSNG ERROR <CR> <LF>				
	067110	020040	020040	020040					
	067116	020040	020040	051040					
	067124	046105	047123	020107					
	067132	042440	051122	051117					
	067140	005015							
9661	067142	042524	052123	021440	.ASCIZ /TEST # ERR PC PORT # PORT # REG ADR GOOD BAD				
	067150	020040	051105	020122					
	067156	041520	020040	047520					
	067164	052122	021440	020040					
	067172	047520	052122	021440					
	067200	020040	042522	020107					
	067206	042101	020122	047507					
	067214	042117	020040	020040					
	067222	040502	000104						
9662	067226	042524	052123	021440	DH11: .ASCIZ TEST # ERR PC PORT # REG ADR CONTENTS				
	067234	020040	051105	020122					

	067242	041520	020040	047520					
	067250	052122	021440	020040					
	067256	042522	020107	042101					
	067264	020122	047503	052116					
	067272	047105	051524	000					
9663	067277	040	020040	020040	DH13:	.ASCII		SEIZE	ERROR CR LF
	067304	020040	020040	020040					
	067312	020040	020040	051440					
	067320	044505	042532	020040					
	067326	042440	051122	051117					
	067334	005015							
9664	067336	042524	052123	021440		.ASCIZ	TEST #	ERR PC	PORT # POP# REG ADR CONTENTS/
	067344	020040	051105	020122					
	067352	041520	020040	047520					
	067360	052122	021440	020040					
	067366	047520	052122	021440					
	067374	020040	042522	020107					
	067402	042101	020122	047503					
	067410	052116	047105	051524					
	067416	000							
9665	067417	040	020040	020040	DH22:	.ASCII /		RELSNG	SEIZE<<CR><<LF>
	067424	020040	020040	020040					
	067432	020040	020040	051040					
	067440	046105	047123	020107					
	067446	051440	044505	042532					
	067454	005015							
9666	067456	042524	052123	021440		.ASCIZ /	TEST #	ERR PC	PORT # PORT #/
	067464	020040	051105	020122					
	067472	041520	020040	047520					
	067500	052122	021440	020040					
	067506	047520	052122	021440					
	067514	000							
9667	067515	040	020040	020040	DH23:	.ASCII /		SEIZE/	<<CR><<LF>
	067522	020040	020040	020040					
	067530	020040	020040	051440					
	067536	044505	042532	005015					
9668	067544	042524	052123	021440		.ASCIZ /	TEST #	ERR PC	PORT # REG ADR CONTENTS
	067552	020040	051105	020122					
	067560	041520	020040	047520					
	067566	052122	021440	020040					
	067574	042522	020107	042101					
	067602	020122	047503	052116					
	067610	047105	051524	000					
9669	067615	040	020040	020040	DH26:	.ASCII /		RELSNG/	<<CR><<LF>
	067622	020040	020040	020040					
	067630	020040	020040	051040					
	067636	046105	047123	0065C7					
	067644	012							
9670	067645	124	051505	020124		.ASCIZ /	TEST #	ERR PC	PORT #/
	067652	020043	042440	051122					
	067660	050040	020103	050040					
	067666	051117	020124	000043					
9671	067674	020040	020040	020040	DH31:	.ASCII		RELSNG	RQSTNG<<CR><<LF>
	067702	020040	020040	020040					
	067710	020040	020040	042522					
	067716	051514	043516	020040					

	067724	050522	052123	043516						
9672	067732	005015								
	067734	042524	052123	021440	.ASCIZ	/TEST	ERR PC	PORT	PORT	
	067742	020040	051105	020122						
	067750	041520	020040	047520						
	067756	052122	021440	020040						
	067764	047520	052122	021440						
	067772	000								
9673	067773	124	051505	020124	DH36:	.ASCIZ	TEST	ERR PC	PORT	
	070000	020043	042440	051122						
	070006	050040	020103	050040						
	070014	051117	020124	000043						
9674	070022	042524	052123	021440	DH42:	.ASCIZ	/TEST	ERR PC		
	070030	020040	051105	020122						
	070036	041520	000							
9675	070041	040	020040	020040	DH44:	.ASCII		RELSNG	ERROR/CR<LF>	
	070046	020040	020040	020040						
	070054	020040	020040	051040						
	070062	046105	047123	020107						
	070070	042440	051122	051117						
	070076	005015								
9676	070100	042524	052123	021440	.ASCIZ	/TEST	ERR PC	PORT	PORT	
	070106	020040	051105	020122						
	070114	041520	020040	047520						
	070122	052122	021440	020040						
	070130	047520	052122	021440						
	070136	000								
9677	070137	040	020040	020040	DH46:	.ASCII		PORT A	PORT B<CR><LF>	
	070144	020040	020040	020040						
	070152	020040	020040	050040						
	070160	051117	020124	020101						
	070166	050040	051117	020124						
	070174	006502	012							
9678	070177	124	051505	020124	.ASCIZ	/TEST	ERR PC	RPDS1	RPDS1	
	070204	020043	042440	051122						
	070212	050040	020103	051040						
	070220	042120	030523	020040						
	070226	051040	042120	030523						
	070234	000								
9679	070235	124	051505	020124	DH55:	.ASCIZ	/TEST	ERR PC	PORT	TIMEOUT VALUE (IN MS)
	070242	020043	042440	051122						
	070250	050040	020103	050040						
	070256	051117	020124	020043						
	070264	052040	046511	047505						
	070272	052125	053040	046101						
	070300	042525	024040	047111						
	070306	046440	024523	000						
9680	070313	044	050122	042101	DH56:	.ASCIZ	/SRPADR			
	070320	000122								
9681										
9682					.EVEN					
9683										
9684	070322	001242	001116	001234	DT1:	.WORD	TSTNUM,\$ERRPC,PTNBR,\$BDADR,\$BDDAT,0			
	070330	001122	001126	000000						
9685	070336	001242	001116	001122	DT3:	.WORD	TSTNUM,\$ERRPC,\$BDADR,\$GDDAT,\$BDDAT,0			
	070344	001124	001126	000000						

9686	070352	001242	001116	001234	DT5:	.WORD	TSTNUM, \$ERRPC, PTNBR, \$BDADR, \$GDDAT, \$BDDAT, 0
	070360	001122	001124	001126			
	070366	000000					
9687	070370	001242	001116	001240	DT6:	.WORD	TSTNUM, \$ERRPC, OPPRT, \$BDADR, \$BDDAT, 0
	070376	001122	001126	000000			
9688	070404	001242	001116	001236	DT7:	.WORD	TSTNUM, \$ERRPC, SEIZPT, PTNBR, \$BDADR, \$GDDAT, \$BDDAT, 0
	070412	001234	001122	001124			
	070420	001126	000000				
9689	070424	001242	001116	001236	DT13:	.WORD	TSTNUM, \$ERRPC, SEIZPT, PTNBR, \$BDADR, \$BDDAT, 0
	070432	001234	001122	001126			
	070440	000000					
9690	070442	001242	001116	001236	DT22:	.WORD	TSTNUM, \$ERRPC, SEIZPT, PTNBR, 0
	070450	001234	000000				
9691	070454	001242	001116	001236	DT23:	.WORD	TSTNUM, \$ERRPC, SEIZPT, \$BDADR, \$BDDAT, 0
	070462	001122	001126	000000			
9692	070470	001242	001116	001236	DT31:	.WORD	TSTNUM, \$ERRPC, SEIZPT, OPPRT, 0
	070476	001240	000000				
9693	070502	001242	001116	001236	DT36:	.WORD	TSTNUM, \$ERRPC, SEIZPT, 0
	070510	000000					
9694	070512	001242	001116	001234	DT37:	.WORD	TSTNUM, \$ERRPC, PTNBR, 0
	070520	000000					
9695	070522	001242	001116	000000	DT42:	.WORD	TSTNUM, \$ERRPC, 0
9696	070530	001242	001116	001170	DT46:	.WORD	TSTNUM, \$ERRPC, \$TMP2, \$TMP3, 0
	070536	001172	000000				
9697	070542	001242	001116	001240	DT54:	.WORD	TSTNUM, \$ERRPC, OPPRT, SEIZPT, 0
	070550	001236	000000				
9698	070554	001242	001116	001236	DT55:	.WORD	TSTNUM, \$ERRPC, SEIZPT, TIME, 0
	070562	001252	000000				
9699	070566	001300	000000		DT56:	.WORD	\$RPADR, 0
9700							
9701	070572	000	000	000	DF1:	.BYTE	0,0,0,0,0
	070575	000	000				
9702	070577	000	000	000	DF5:	.BYTE	0,0,0,0,0,0
	070602	000	000	000			
9703	070605	000	000	000	DF7:	.BYTE	0,0,0,0,0,0,0
	070610	000	000	000			
	070613	000					
9704	070614	000	000	000	DF31:	.BYTE	0,0,0,0
	070617	000					
9705	070620	000	000	000	DF36:	.BYTE	0,0,0
9706	070623	000	000		DF42:	.BYTE	0,0
9707	070625	000	000	000	DF55:	.BYTE	0,0,0,1
	070630	001					
9708	070631	000			DF56:	.BYTE	0
9709							
9710							.EVEN
9711							
9712							
9713							::*****
9714							
9715							.SBTTL CONSTANTS, TABLES, ETC
9716							::*****
9717							
9718							:TABLE OF TEST STARTING ADDRESSES
9719							
9720							

9721	070632	003120	TSTADR: .WORD	TST1	: STARTING ADDRESS OF TEST	1
9724	070634	004520	.WORD	TST2	: STARTING ADDRESS OF TEST	2
(1)	070636	006120	.WORD	TST3	: STARTING ADDRESS OF TEST	3
(1)	070640	007520	.WORD	TST4	: STARTING ADDRESS OF TEST	4
(1)	070642	010656	.WORD	TST5	: STARTING ADDRESS OF TEST	5
(1)	070644	012014	.WORD	TST6	: STARTING ADDRESS OF TEST	6
(1)	070646	013004	.WORD	TST7	: STARTING ADDRESS OF TEST	7
(1)	070650	013774	.WORD	TST10	: STARTING ADDRESS OF TEST	10
(1)	070652	014416	.WORD	TST11	: STARTING ADDRESS OF TEST	11
(1)	070554	015040	.WORD	TST12	: STARTING ADDRESS OF TEST	12
(1)	070656	016416	.WORD	TST13	: STARTING ADDRESS OF TEST	13
(1)	070660	017774	.WORD	TST14	: STARTING ADDRESS OF TEST	14
(1)	070662	021276	.WORD	TST15	: STARTING ADDRESS OF TEST	15
(1)	070664	022600	.WORD	TST16	: STARTING ADDRESS OF TEST	16
(1)	070666	024040	.WORD	TST17	: STARTING ADDRESS OF TEST	17
(1)	070670	024726	.WORD	TST20	: STARTING ADDRESS OF TEST	20
(1)	070672	025614	.WORD	TST21	: STARTING ADDRESS OF TEST	21
(1)	070674	026774	.WORD	TST22	: STARTING ADDRESS OF TEST	22
(1)	070676	030154	.WORD	TST23	: STARTING ADDRESS OF TEST	23
(1)	070700	031164	.WORD	TST24	: STARTING ADDRESS OF TEST	24
9727	070702	032174	.WORD	TST25	: STARTING ADDRESS OF TEST	25
(1)	070704	033402	.WORD	TST26	: STARTING ADDRESS OF TEST	26
(1)	070706	034610	.WORD	TST27	: STARTING ADDRESS OF TEST	27
(1)	070710	036604	.WORD	TST30	: STARTING ADDRESS OF TEST	30
(1)	070712	037300	.WORD	TST31	: STARTING ADDRESS OF TEST	31
(1)	070714	040540	.WORD	TST32	: STARTING ADDRESS OF TEST	32
(1)	070716	042000	.WORD	TST33	: STARTING ADDRESS OF TEST	33
(1)	070720	043070	.WORD	TST34	: STARTING ADDRESS OF TEST	34
(1)	070722	044160	.WORD	TST35	: STARTING ADDRESS OF TEST	35
(1)	070724	044774	.WORD	TST36	: STARTING ADDRESS OF TEST	36
9730	070726	045610	.WORD	TST37	: STARTING ADDRESS OF TEST	37
(1)	070730	046572	.WORD	TST40	: STARTING ADDRESS OF TEST	40
(1)	070732	047554	.WORD	TST41	: STARTING ADDRESS OF TEST	41
(1)	070734	051162	.WORD	TST42	: STARTING ADDRESS OF TEST	42
(1)	070736	052570	.WORD	TST43	: STARTING ADDRESS OF TEST	43
(1)	070740	053452	.WORD	TST44	: STARTING ADDRESS OF TEST	44
(1)	070742	054240	.WORD	TST45	: STARTING ADDRESS OF TEST	45
(1)	070744	055306	.WORD	TST46	: STARTING ADDRESS OF TEST	46

:ATTENTION BIT TABLE

9731						
9732						
9733						
9734	070746	001	ATABIT: .BYTE	1	: ATTENTION BIT FOR DRIVE	0
9735	070747	002	.BYTE	2	: ATTENTION BIT FOR DRIVE	1
9736	070750	004	.BYTE	4	: ATTENTION BIT FOR DRIVE	2
9737	070751	010	.BYTE	10	: ATTENTION BIT FOR DRIVE	3
9738	070752	020	.BYTE	20	: ATTENTION BIT FOR DRIVE	4
9739	070753	040	.BYTE	40	: ATTENTION BIT FOR DRIVE	5
9740	070754	100	.BYTE	100	: ATTENTION BIT FOR DRIVE	6
9741	070755	200	.BYTE	200	: ATTENTION BIT FOR DRIVE	7
9742						
9743	070756	000046	MAXTN: .WORD	\$TN-1	: MAXIMUM TEST NUMBER	
9744						
9745		000001	.END			

ABS	=	000200	CKCLK3	=	056616	DT01	=	000002	EM36	=	065142	KYBCTL	=	001274
ACL	=	000040	CKERR	=	001244	DT02	=	000004	EM37	=	065214	LF	=	000012
ACU	=	100000	CKSWR	=	104407	DT03	=	000010	EM4	=	062653	LST	=	002000
ADDRIS	=	062414	CLOCK	=	056626	DT04	=	000020	EM40	=	065262	MAXTN	=	070756
ADRERR	=	062167	CLR	=	000040	DT05	=	000040	EM41	=	065337	MCLK	=	000002
AJE	=	001000	CR	=	000015	DT06	=	000100	EM42	=	065401	MCPE	=	020000
ASR1	=	001232	CRLF	=	000200	DT07	=	000200	EM43	=	065467	MHS	=	001000
ATA	=	100000	CSF	=	000002	DT08	=	000400	EM44	=	065544	MINX	=	000004
ATABIT	=	070746	CSU	=	000010	DT1	=	070322	EM45	=	065621	MOH	=	020000
ATO	=	000001	CK	=	100000	DT13	=	070424	EM46	=	065700	MOL	=	010000
AT1	=	000002	DCL	=	000100	DT22	=	070442	EM47	=	065747	MPE	=	000400
AT2	=	000004	DCU	=	000001	DT23	=	070454	EM5	=	062704	MRD	=	000020
AT3	=	000010	DDISP	=	177570	DT3	=	070336	EM50	=	066035	MSE	=	000020
AT4	=	000020	DE1	=	000040	DT31	=	070470	EM51	=	066120	MSTCK	=	000010
AT5	=	000040	DF20	=	000002	DT36	=	070502	EM52	=	066203	MWR	=	000040
AT6	=	000100	DF1	=	070572	DT37	=	070512	EM53	=	066301	MXF	=	001000
AT7	=	000200	DF31	=	070614	DT42	=	070522	EM54	=	066362	NBA	=	100000
A16	=	000400	DF36	=	070620	DT46	=	070530	EM55	=	066455	NED	=	010000
A17	=	001000	DF42	=	070623	DT5	=	070352	EM56	=	066522	NEM	=	004000
BADNO	=	062354	DF5	=	070577	DT54	=	070542	EM6	=	062752	NHS	=	002000
BAI	=	000010	DF55	=	070625	DT55	=	070554	EM7	=	063052	NOATA	=	000001
BIT0	=	000001	DF56	=	070631	DT56	=	070566	ENTERA	=	062137	NOCLOC	=	062265
BIT00	=	000001	DF7	=	070605	DT6	=	070370	ERR	=	040000	NOSEIZ	=	001246
BIT01	=	000002	DH1	=	066564	DT7	=	070404	ERRVEC	=	000004	NTRH11	=	062474
BIT02	=	000004	DH11	=	067226	DVA	=	004000	EXEC	=	002622	OCYL	=	100000
BIT03	=	000010	DH13	=	067277	ECH	=	000100	EXT1	=	000001	OFREV	=	000200
BIT04	=	000020	DH22	=	067417	ECI	=	004000	EXT10	=	000010	OF100	=	000004
BIT05	=	000040	DH23	=	067515	EMTVEC	=	000030	EXT2	=	000002	OF200	=	000010
BIT06	=	000100	DH26	=	067615	EM1	=	062526	EXT20	=	000020	OF25	=	000001
BIT07	=	000200	DH3	=	066635	EM10	=	063133	EXT4	=	000004	OF400	=	000020
BIT08	=	000400	DH31	=	067674	EM11	=	063163	EXT40	=	000040	OF50	=	000002
BIT09	=	001000	DH36	=	067773	EM12	=	063247	FEN	=	000200	OF800	=	000040
BIT1	=	000002	DH4	=	066704	EM13	=	063317	FER	=	000020	OPE	=	020000
BIT10	=	002000	DH42	=	070022	EM14	=	063404	FMT22	=	010000	OPI	=	020000
BIT11	=	004000	DH44	=	070041	EM15	=	063447	F1	=	000002	OPPR	=	001240
BIT12	=	010000	DH46	=	070137	EM16	=	063526	F2	=	000004	OR	=	000200
BIT13	=	020000	DH5	=	067027	EM17	=	063601	F3	=	000010	PAR	=	000010
BIT14	=	040000	DH55	=	070235	EM2	=	062547	F4	=	000020	PAT	=	000020
BIT15	=	100000	DH56	=	070313	EM20	=	063660	F5	=	000040	PGE	=	002000
BIT2	=	000004	DH7	=	067103	EM21	=	063740	GO	=	000001	PGM	=	001000
BIT3	=	000010	DIG8	=	000004	EM22	=	064013	GRV	=	000010	PIP	=	020000
BIT4	=	000020	DISPLA	=	001142	EM23	=	064100	GTSWR	=	104406	PIRQ	=	177772
BIT5	=	000040	DISPRE	=	000174	EM24	=	064146	HCE	=	000200	PIRQVE	=	000240
BIT6	=	000100	DLT	=	100000	EM25	=	064225	HCI	=	002000	PLU	=	020000
BIT7	=	000200	DL64	=	000020	EM26	=	064270	HCRC	=	000400	PORTA	=	001224
BIT8	=	000400	DMO	=	000001	EM27	=	064355	HT	=	000011	PORTAI	=	062211
BIT9	=	001000	DPR	=	000400	EM3	=	062571	IAE	=	002000	PORTB	=	001226
BPTVEC	=	000014	DRQ	=	004000	EM30	=	064433	IE	=	000100	PORTBI	=	062237
CHANGE	=	003004	DRY	=	000200	EM31	=	064530	ILF	=	000001	PORTC	=	001230
CHGADR	=	001276	DSWR	=	177570	EM32	=	064605	ILR	=	000002	PRE	=	000020
CKCLK	=	056474	DTE	=	010000	EM33	=	064656	IOTVEC	=	000020	PRO	=	000000
CKCLK1	=	056544	DTSY	=	000200	EM34	=	064760	IR	=	000100	PR1	=	000040
CKCLK2	=	056606	DT00	=	000001	EM35	=	065063	IXE	=	004000	PR2	=	000100

PR3 = 000140
 PR4 = 000200
 PR5 = 000240
 PR6 = 000300
 PR7 = 000340
 PS = 177776
 PSEL = 002000
 PSU = 000001
 PSW = 177776
 PTNBR = 001234
 PWRVEC = 000024
 RAW = 000020
 ROCHR = 104410
 ROLIN = 104411
 RDOCT = 104412
 ROY = 000200
 RELERR = 001250
 RELOK = 000001
 RESREG = 104414
 RESVEC = 000010
 RMR = 000004
 RPAS = 000016
 RPBA = 000004
 RPCA = 000034
 RPCC = 000036
 RPCS1 = 000000
 RPCS2 = 000010
 RPDA = 000006
 RPDB = 000022
 RPDS1 = 000012
 RPDT = 000026
 RPEC1 = 000044
 RPEC2 = 000046
 RPER1 = 000014
 RPER2 = 000040
 RPER3 = 000042
 RPLA = 000020
 RPMR = 000024
 RPOF = 000032
 RPSN = 000030
 RPWC = 000002
 RE = %000006
 R7 = %000007
 SAVREG = 104413
 SC = 100000
 SC1 = 000100
 SC10 = 001000
 SC2 = 000200
 SC20 = 002000
 SC4 = 000400
 SEIZPT = 001236
 SKI = 040000
 STACK = 001100

START = 002064
 START1 = 002072
 START2 = 002100
 STKLMT = 177774
 SWR = 001140
 SWREG = 000176
 SWO = 000001
 SW00 = 000001
 SW01 = 000002
 SW02 = 000004
 SW03 = 000010
 SW04 = 000020
 SW05 = 000040
 SW06 = 000100
 SW07 = 000200
 SW08 = 000400
 SW09 = 001000
 SW1 = 000002
 SW10 = 002000
 SW11 = 004000
 SW12 = 010000
 SW13 = 020000
 SW14 = 040000
 SW15 = 100000
 SW2 = 000004
 SW3 = 000010
 SW4 = 000020
 SW5 = 000040
 SW6 = 000100
 SW7 = 000200
 SW8 = 000400
 SW9 = 001000
 TAP = 040000
 TBITVE = 000014
 TDF = 000040
 TESTNO = 062334
 TEST1 = 003172
 TEST10 = 014046
 TEST11 = 014470
 TEST12 = 015112
 TEST13 = 016470
 TEST14 = 020046
 TEST15 = 021350
 TEST16 = 022652
 TEST17 = 024112
 TEST2 = 004572
 TEST20 = 025000
 TEST21 = 025666
 TEST22 = 027046
 TEST23 = 030226
 TEST24 = 031236
 TEST25 = 032246
 TEST26 = 033454

TEST27 = 034662
 TEST3 = 006172
 TEST30 = 036856
 TEST31 = 037352
 TEST32 = 040612
 TEST33 = 042052
 TEST34 = 043142
 TEST35 = 044232
 TEST36 = 045046
 TEST37 = 045662
 TEST4 = 007572
 TEST40 = 046644
 TEST41 = 047626
 TEST42 = 051234
 TEST43 = 052642
 TEST44 = 053524
 TEST45 = 054412
 TEST46 = 055360
 TEST5 = 010730
 TEST6 = 012066
 TEST7 = 013056
 TIME = 001252
 TIMEA = 001256
 TIMEAM = 001262
 TIMEAP = 001260
 TIMEB = 001264
 TIMEBM = 001270
 TIMEBP = 001266
 TIMES = 001272
 TITLE = 062040
 TKVEC = 000060
 TOLER = 056660
 TPVEC = 000064
 TRAPVE = 000034
 TRE = 040000
 TRK1 = 004000
 TRK10 = 040000
 TRK2 = 010000
 TRK20 = 100000
 TRK4 = 020000
 TRTVEC = 000014
 TSTADR = 070532
 TSTERR = 062402
 TSTNUM = 001242
 TST1 = 003120
 TST1AA = 003114
 TST10 = 013774
 TST11 = 014416
 TST12 = 015040
 TST13 = 016416
 TST14 = 017774
 TST15 = 021276
 TST16 = 022600

TST17 = 024040
 TST2 = 004520
 TST20 = 024726
 TST21 = 025614
 TST22 = 026774
 TST23 = 030154
 TST24 = 031164
 TST25 = 032174
 TST26 = 033402
 TST27 = 034610
 TST3 = 006120
 TST30 = 036604
 TST31 = 037300
 TST32 = 040540
 TST33 = 042000
 TST34 = 043070
 TST35 = 044160
 TST36 = 044774
 TST37 = 045610
 TST4 = 007520
 TST40 = 046572
 TST41 = 047554
 TST41B = 050630
 TST42 = 051162
 TST42B = 052236
 TST43 = 052570
 TST44 = 053452
 TST45 = 054340
 TST46 = 055306
 TST5 = 010656
 TST6 = 012014
 TST7 = 013004
 TUF = 000100
 TYPDS = 104405
 TYPE = 104401
 TYPOC = 104402
 TYPON = 104404
 TYPOS = 104403
 UNS = 040000
 UPE = 020000
 US1 = 000001
 US2 = 000002
 JS4 = 000004
 UWR = 000010
 VUF = 000002
 VU30 = 010000
 JV = 000100
 VVSET = 000001
 WAO = 000002
 WATCH = 001254
 WCE = 040000
 WCF = 000040
 WCU = 000001

WLE = 004000
 WPL = 004000
 WPU = 000400
 WSU = 000004
 \$AUTOB = 001134
 \$BDAOR = 001122
 \$BDDAT = 001126
 \$BELL = 001202
 \$CHARC = 057674
 \$CKSWR = 060624
 \$CMTAG = 001100
 \$CM1 = 000001
 \$CM2 = 000002
 \$CM3 = 000001
 \$CM4 = 000005
 \$CNTLG = 061466
 \$CNTLU = 061461
 \$CRLF = 001207
 \$DBLK = 060342
 \$DOAGN = 056464
 \$DTBL = 060332
 \$ENDAD = 056454
 \$ENDCT = 056320
 \$ENULL = 056470
 \$EOP = 056254
 \$EOPCT = 056312
 \$ERFLG = 001103
 \$ERMAX = 001115
 \$ERROR = 057152
 \$ERRPC = 001116
 \$ERRTB = 001304
 \$ERRTY = 057304
 \$ERTTL = 001112
 \$ESCAP = 001200
 \$FILLC = 001156
 \$FILLS = 001155
 \$GDADR = 001120
 \$GODAT = 001124
 \$GET42 = 056444
 \$GTSWR = 060714
 \$HO = 000000
 \$HIOCT = 061654
 \$ICNT = 001104
 \$INTAG = 001135
 \$ITEMB = 001114
 \$LF = 001210
 \$LKCSB = 001214
 \$LKCSR = 001212
 \$LKS = 001220
 \$LLVEC = 001222
 \$LPADR = 001106
 \$LPERR = 001110
 \$LPVEC = 001216

\$MNEW 061504
 \$MSWR 061473
 \$MXCNT 057150
 \$NULL 001154
 \$NWTST= 000001
 \$OCNT 060122
 \$OMODE 060124
 \$OVER 057134
 \$PASS 001100
 \$QUES 001206
 \$RDCHR 061126
 \$RDLIN 061216
 \$RDOCT 061516

\$RDSZ = 000007
 \$REGAD 001160
 \$REGO 001162
 \$RESRE 061714
 \$RPADR 001300
 \$RPVEC 001302
 \$RTNAD 056466
 \$SAVRE 061656
 \$SCOPE 056720
 \$SETUP= 000127
 \$STUP = 177777
 \$SVLAD 057106
 \$SVPC = 000200

\$SWR = 166000
 \$SWRMK= 000000
 \$TIMES 001176
 \$TKB 001146
 \$TKCNT 060352
 \$TKINT 060362
 \$TKQEN= 060361
 \$TKQIN 060354
 \$TKQOU 060356
 \$TKQSR 060360
 \$TKS 001144
 \$TKSRV 060432
 \$TMPD 001164

\$TMP1 001166
 \$TMP2 001170
 \$TMP3 001172
 \$TMP4 001174
 \$TN = 000047
 \$TPB 001152
 \$TPFLG 001157
 \$TPS 001150
 \$TRAP 061752
 \$TRAP2 061774
 \$TRP = 000015
 \$TRPAD 062006
 \$TSTNM 001102

\$TTYIN 061452
 \$TYPDS 060126
 \$TYPE 057460
 \$TYPEC 057630
 \$TYPEX 057676
 \$TYPOC 057724
 \$TYPON 057740
 \$TYPDS 057700
 \$TSTR 056732
 \$\$GET4= 000000
 \$OFILL 060123
 = 070760

. ABS. 070760 000

ERRORS DETECTED: 0

CZRJEC, CZRJEC.SEQ=CZRJEC.SML, CZRJEC.P11
 RUN-TIME: 30 42 .6 SECONDS
 RUN-TIME RATIO: 740.73=10.0
 CORE USED: 40K (79 PAGES)

H15