

# RH11,RS03/4

BASIC FUNCTION DIAGNOSTIC  
CZRSBG0

AH-9315G-MC

COPYRIGHT © 73-78

FICHE 1 OF 1

JAN 1979

**digital**

MADE IN USA

1  
2  
3

.REM @

IDENTIFICATION  
-----

PRODUCT CODE: AC-9314G-MC  
PRODUCT NAME: CZRSBG0 RH11-RS03/LA04 BASIC FUNCTION  
DIAGNOSTIC  
DATE CREATED: DEC 1978  
MAINTAINER: DIAGNOSTIC GROUP

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.

THE SOFTWARE DESCRIBED IN THIS DOCUMENT IS FURNISHED UNDER A LICENSE AND MAY ONLY BE USED OR COPIED IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE.

DIGITAL EQUIPMENT CORPORATION ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS SOFTWARE ON EQUIPMENT THAT IS NOT SUPPLIED BY DIGITAL.

COPYRIGHT (C) 1973,1978 BY DIGITAL EQUIPMENT CORPORATION

CZRSB-G RH11-RS03/LA-RS04 BASIC FUNCTION DIAGNOSTIC PAGE 2  
TABLE OF CONTENTS

CONTENTS

1. ABSTRACT
2. REQUIREMENTS
- 2.1 EQUIPMENT
- 2.3 PRELIMINARY PROGRAMS
3. LOADING PROCEDURE
4. STARTING PROCEDURE
- 4.1 CONTROL SWITCH SETTINGS
- 4.2 STARTING ADDRESS
- 4.3 PROGRAM AND/OR OPERATING PROCEDURE
5. OPERATIONAL SWITCH SETTINGS
- 5.2 SUBROUTINE ABSTRACT
6. ERRORS
7. RESTRICTIONS
8. MISCELLANEOUS
- 8.1 EXECUTION TIME
- 8.2 STACK POINTER
9. WRITE LOCK TEST
10. TEST DESCRIPTION

CZRSB-G RH11-RS03/LA-RS04 BASIC FUNCTION DIAGNOSTIC PAGE 3  
DESCRIPTION

1. ABSTRACT

THIS DIAGNOSTIC WAS DESIGNED TO TEST RS03,RS03/LA AND RS04 DRIVES.

THIS IS A BASIC FUNCTION DIAGNOSTIC WHICH IS USED TO VERIFY THAT THE (RH11) CONTROLLER AND THE (RS03,RS03/LA OR RS04) DISKS ARE OPERATING CORRECTLY. THIS IS NOT A RELIABILITY DIAGNOSTIC AND THEREFORE SHOULD NOT BE USED AS ONE. THIS PROGRAM CAN TEST UP TO 8 DRIVES. THE DRIVES CAN BE INTERMIXED AND IN ANY ORDER.

IF THE OPERATOR WOULD LIKE TO CHECK THE DISK REGISTERS PRIOR TO ENTERING THIS DIAGNOSTIC, THERE ARE SOME ROUTINES IN THE BACK OF THE DIAGNOSTIC WHICH CAN BE USED. THESE ROUTINES WILL ALLOW THE OPERATOR TO LOAD THE REGISTERS THROUGH THE SWITCHES. PLEASE REFERENCE THE STARTING ADDRESSES THAT WILL TEST THE REGISTERS YOU DESIRE.

2. REQUIREMENTS

2.1 EQUIPMENT

PDP11 STANDARD COMPUTER WITH A MINIMUM OK 8K OF MEMORY, AND AN RH11 CONTROLLER WITH A RS03, RS03/LA OR RS04 DISK.

2.3 PRELIMINARY PROGRAMS

NONE

3. LOADING PROCEDURE

USE STANDARD PROCEDURE FOR ABS TAPES.

4. STARTING PROCEDURE

4.1 CONTROL SWITCH SETTINGS

SEE 5.1.1 (ALL DOWN FOR WORST CASE TESTING)

4.2 STARTING ADDRESS

CZRSB-G RH11-RS03/LA-RS04 BASIC FUNCTION DIAGNOSTIC PAGE 4  
DESCRIPTION

#### 4.3 PROGRAM AND/OR OPERATOR ACTION

LOAD PROGRAM INTO MEMORY USING ABS LOADER.

##### 1. STARTING ADDRESS 200.

- A. SET SWITCHES (SEE SEC 5.1.1) ALL DOWN FOR WORST CASE. IF SWITCHLESS CPU, SIMPLY
- B. PRESS START.
- C. THE PROGRAM WILL LOOP AND BELL WILL RING ONCE EVERY PASS
- D. THE DISPLAY ON THE 11/45 WILL SHOW THE ITERATION COUNT IN THE LEFT BYTE AND TEST NUMBER IN THE RIGHT. TO USE, SET THE DATA DISPLAY SWITCH TO THE DISPLAY POSITION.
- E. THE PROGRAM WILL TEST ALL RS03, RS03/LA AND RS04 DISKS.

##### 2. STARTING ADDRESSES FOR TESTING THE RH11-RS03/LA/04 REGISTERS USING THE SWITCH REGISTER. ON SWITCH-LESS MACHINES THESE ROUTINES ARE USEFULL FOR SCOPING. SIMPLY STRIKE ^G ANYTIME AFTER PRESSING START TO ENTER OR CHANGE VALUE DESIRED.

- A. 250 WORD COUNT REGISTER TEST
- B. 254 BUS ADDRESS REG. TEST
- C. 260 DISK ADDRESS REG. TEST
- D. 264 DRIVE STATUS REG. TEST
- E. 270 ERROR REG. TEST
- F. 274 LOOK AHEAD REG. TEST
- G. 300 RSCS2 REG. TEST
- H. 304 ATTENTION SUMMARY REG. TEST
- I. 310 MAINTENANCE REG. TEST
- J. 314 RSCS1 REG TEST

##### 5. OPERATIONAL SWITCH SETTINGS

THIS PROGRAM HAS BEEN MODIFIED TO RUN ON A PROCESSOR WITH OR WITHOUT A HARDWARE SWITCH REGISTER. WHEN FIRST EXECUTED THE PROGRAM TESTS THE EXISTENCE OF A HARDWARE SWITCH REGISTER. IF NOT FOUND A SOFTWARE SWITCH REGISTER LOCATION (SWREG=LOC.176) IS DEFAULTED TO. IF THIS IS THE CASE, UPON EXECUTION THE CONTENTS OF THE SWREG ARE DUMPED IN OCTAL ON THE CONSOLE TTY AND ANY CHANGES ARE REQUESTED

(I.E.) SWR=XXXXXX NEW=

POSSIBLE RESPONSES ARE:

- 1. <CR> IF NO CHANGES ARE TO BE MADE.
- 2. 6 DIGITS 0-7 TO REPRESENT IN OCTAL THE NEW SWITCH REGISTER VALUE; LAST DIGIT FOLLOWED BY <CR>.
- 3. ^U TO ALLOW REENTERING VALUE IF ERROR IS COMMITTED

KEYING IN SWREG VALUE.

CZRSB-G RH11-RS03/LA-RS04 BASIC FUNCTION DIAGNOSTIC PAGE 5  
DESCRIPTION

BUILT INTO THE PROGRAM IS THE ABILITY TO DYNAMICALLY CHANGE THE CONTENTS OF SWREG DURING PROGRAM EXECUTION. BY STRIKING ^G (CNTRL G) ON CONSOLE TTY THE OPERATOR SETS A REQUEST FLAG TO CHANGE THE CONTENTS OF SWREG, WHICH IS PROCESSED IN KEY AREAS OF THE PROGRAM CODE (IE) ERROR ROUTINES, AFTER HALTS END OF PASS, AND OTHER APPLICABLE AREAS.

SWITCH SETTINGS ARE:

SW<15> = 1 ..... HALT ON ERROR  
SW<14> = 1 ..... LOOP ON TEST  
SW<13> = 1 ..... INHIBIT TYPEOUTS  
SW<12> = 1 ..... INHIBIT OBUFSV FROM CHANGING WHEN LOOKING  
FOR MEMORY ON -B- PORT  
SW<11> = 1 ..... INHIBIT ITERATIONS OF SUBTEST  
SW<10> = 1 ..... BELL ON ERROR  
0 ..... BELL ON PASS COMPLETE  
SW<09> = 1 ..... LOOP ON ERROR  
SW<08> = 1 ..... LOOP ON TEST IN SW<7:0>

## 5.2 SUBROUTINE ABSTRACTS

### 5.2.1 SCOPE

THIS SUBROUTINE CALL IS PLACED BETWEEN EACH SUBTEST IN THE INSTRUCTION SECTION. IT RECORDS THE STARTING ADDRESS OF EACH SUBTEST AS IT IS BEING ENTERED IN LOCATION 'LAD'. IF A SCOPE LOOP IS REQUESTED, THE CURRENT SUBTEST WILL BE LOOPED UPON. SW<11> ON A 1 INHIBITS ITERATION OF SUBTESTS. THE CONTENTS OF LAD MAY BE USED TO DETERMINE THE LAST SUBTEST SUCCESSFULLY COMPLETED.

### 5.2.2 HLT

THIS ROUTINE PRINTS OUT AN ERROR MESSAGE (SEE 6.1). TO INHIBIT TYPEOUTS, PUT SW<13> ON A 1.

### 5.2.3 TRAPCATCHER

A ".+2" - "HALT" SEQUENCE IS REPEATED FROM 0 - 776 TO CATCH ANY UNEXPECTED TRAPS. THUS ANY UNEXPECTED TRAPS OR INTERRUPTS WILL HALT AT THE VECTOR + 2.

## 6. ERRORS

CZRSB-G RH11-RS03/LA-RS04 BASIC FUNCTION DIAGNOSTIC PAGE 6  
DESCRIPTION

6.1 ERROR PRINTOUT

THE FORMAT IS AS FOLLOWS:

ADR CS1 = ----- CS2 = ----- ER = -----  
GOOD = ----- BAD = -----

WHERE:

CS1,CS2,ER ETC. = RS11 DISK REGISTERS.  
GOOD = EXPECTED DATA.  
BAD = DATA RECEIVED.

TO FIND THE FAILING TEST, LOOK AT THE LISTING ABOVE THE ADDRESS TYPED.

6.2 ERROR RECOVERY

RESTART AT 200

7. RESTRICTIONS

NONE

8. MISCELLANEOUS

8.1 EXECUTION TIME

A BELL WILL RING WITHIN 1 MINUTE WITH ALL SWITCHES DOWN.

8.2 STACK POINTER

STACK IS INITIALLY SET TO 500

9. WRITE LOCK TEST

THE WRITE LOCK TEST REQUIRES OPERATOR INTERVENTION. THE STARTING ADDRESS FOR THIS TEST IS 220. THE PROGRAM WILL TELL THE OPERATOR WHICH SWITCHES HAVE TO BE SET.



10. TEST DESCRIPTION

CZRSB-G RH11-RS03/LA-RS04 BASIC FUNCTION DIAGNOSTIC PAGE 7  
DESCRIPTION

1. TEST RSCS2  
CLEAR ALL READ/WRITE BITS AND CHECK. SET ALL R/W BITS AND CHECK. NOW CLEAR AND RECHECK.
2. TEST FOR ONLINE DRIVES  
SET ERROR BITS IN RSER. THIS CAUSES ATTENTION SUMMARY BITS TO SET IN RSAS. DO FOR ALL DRIVES. RSAS HAS NOT YET BEEN TESTED. SO IN THE CASE OF NO BITS IN RSAS SETTING, DRIVE 0 IS TESTED.
3. RESET TEST FOR REGISTERS  
SET ALL R/W BITS IN RSCS1, RSCS2, RSBA, RSDA, RSER, RSWC, RSDB, AND RSMR. DO A RESET AND TEST ALL R/W BITS TO BE CLEARED.
4. SET AND CLEAR ALL REGISTERS  
SET ALL R/W BITS IN RSCS1, RSCS2, RSBA, RSDA, RSER, RSWC, RSDB AND RSMR AND TEST. SET ALTERNATE BITS AND CHECK TO MAKE SURE BITS ARE NOT TIED TOGETHER. NOW SET ALL BITS AND CLEAR THEM TO MAKE SURE ALL CAN BE CLEARED ONCE SET.
5. RANDOM NUMBER TEST FOR RSWC AND RSDA  
THIS TEST GENERATES RANDOM NUMBERS AND LOADS THEM INTO RSWC, RSDA AND RSBA.
6. TEST "CLEAR BIT" IN RSCS2  
SET ALL R/W BITS IN RSCS1, RSCS2, RSBA, RSDA, RSER, RSWC, RSDB, AND RSMR. SET CLEAR BIT IN RSCS2. NOW TEST ALL R/W BITS FOR 0 IN ALL THE ABOVE REGISTERS.
7. TEST DLT AND TRE BITS  
DO A READ FROM THE SILO. THIS SHOULD CAUSE A DLT AND A TRE ERROR BECAUSE THE SILO IS EMPTY.
8. CLEAR DLT AND TRE  
CLEAR BY SETTING TRE IN RSCS1 AND TEST.
9. LOAD RSDB WITH ALL ONES AND ALL ZEROS  
LOAD RSDB WITH A WORD OF ZEROS AND A WORD OF ONES. WAIT FOR "OR" TO SET AND THEN CHECK OUTPUT OF SILO. IF OR DID NOT SET ERROR MESSAGE APPEARS.
10. TEST FOR 66 LOCATIONS IN SILO

THIS IS DONE BY PUTTING A BINARY COUNT IN EVERY LOCATION AND

CZRSB-G RH11-RS03/LA-RS04 BASIC FUNCTION DIAGNOSTIC PAGE 8  
DESCRIPTION

CHECKING THE OUTPUT FOR 66 WORDS.

11. TEST DLT ERROR

THIS IS DONE BY LOADING THE SILO WITH 67 WORDS WITHOUT READING ANY OUT. THIS SHOULD CAUSE DLT TO SET.

12. FLOAT A '1' AND A '0' THROUGH THE SILO

LOAD THE SILO WITH A WORD OF ZEROS AND FLOAT A '1' THROUGH THE WORD. THEN LOAD THE SILO WITH A WORD OF ALL ONES AND FLOAT A '0' THROUGH THE WORD. CHECK THE OUTPUT OF THE SILO FOR THE CORRECT ANSWER.

13. TEST NO-OP FUNCTION

THE NO-OP FUNCTION IS TESTED WITH AND WITHOUT ERROR BITS SET. ALL THE REGISTERS ARE CHECKED AFTER BOTH CASES.

14. TEST DRIVE CLEAR FUNCTION

FIRST SET ALL R/W BITS IN RSDA, RSWC, RSER, AND RSMR. DO A DRIVE CLEAR FUNCTION. NOW TEST ALL REGISTERS FOR CORRECT DATA.

15. EXECUTE A ONE WORD WRITE FUNCTION

SET RSWC TO -1. MOV -1 INTO OUTBUF. LOAD RSBA WITH OUTBUF. DO A WRITE TEST RDY BIT FOR 0 THEN WAIT FOR IT TO SET. TIME OUT TO ERROR IF RDY BIT DOESN'T SET AND CHECK FOR ERROR CONDITIONS. TEST RSDA FOR CORRECT ADDRESS. TEST WORD COUNT FOR 0. THIS IS TESTED ON -A- AND -B- PORT.

16. EXECUTE A ONE WORD WRITE CHECK

SET UP RSDA, RSBA, RSWC AND OUTBUF AS IN THE WRITE FUNCTION TEST. DO A WRITE CHECK FUNCTION. TEST RDY AS DONE IN THE WRITE TEST. CHECK FOR WRITE CHECK ERROR. THEN TEST RSDA, RSWC AND RSBA FOR CORRECT DATA. THIS IS TESTED ON -A- AND -B- PORT.

17. TEST READ FUNCTION

SETUP RSDA, RSBA, RSWC AND OUTBUF AS IN THE WRITE FUNCTION DO A READ FUNCTION. TEST RDY BIT AS DONE IN THE WRITE FUNCTION. TEST FOR ERRORS. ALSO TEST RSDA, RSWC AND RSBA FOR CORRECT DATA. THIS IS TESTED ON -A- AND -B- PORT.

18. TEST BLOCK SEARCH FUNCTION, PIP AND DRY BITS AND ADDR. CONF. BIT

DO A BLOCK SEARCH FOR SECTOR 32, LOOP ON ADDR. CONF. BIT IN

RSMR. IF IT DOESN'T SET, TIMEOUT. WHEN YOU GET THERE DO A  
BLOCK SEARCH FOR SECTOR 0. NOW WE KNOW THAT WE HAVE TIME TO

CZRSB-G RH11-RS03/LA-RS04 BASIC FUNCTION DIAGNOSTIC PAGE 9  
DESCRIPTION

TEST FOR DRY AND PIP BITS BEFORE FINDING SECTOR 0. FOR PIP SHOULD SET AND DRY SHOULD CLEAR BEFORE FINDING SECTOR 0. ONCE SECTOR 0 IS FOUND PIP SHOULD CLEAR AND DRY SHOULD SET. IF DRY DOES NOT SET A TIME OUT ERROR WILL OCCUR INDICATING SECTOR 0 WAS NOT FOUND. SC IN RSCS1 SHOULD ALSO SET. RSBA AND RSWC SHOULD NOT MOVE, THIS IS ALSO TESTED.

19. ILLEGAL FUNCTION CODE TEST

IN THIS TEST RSBA, RSWC AND RSDA ARE SET UP AS IF TO DO A LEGAL FUNCTION. AN ILLEGAL FUNCTION IS THEN EXECUTED. THE PROGRAM TEST FOR ILF AND ERR BITS TO SET. RSBA, RSWC AND RSDA ARE ALSO TESTED FOR CORRECT DATA. THIS IS DONE FOR ALL THE ILLEGAL FUNCTIONS.

FOR AN AID IN TROUBLE SHOOTING THE ILLEGAL FUNCTION CODE CAN BE LOADED INTO LOCATION ILLTAB OR ILFTB2, DEPENDING ON WHICH ILLEGAL FUNCTION TEST YOU WISH TO LOOP ON. IN THE NEXT LOCATION, FOLLOWING THE ILLEGAL FUNCTION, A 0 MUST BE LOADED. NOW BY SETTING SWITCH 14 (LOOP ON TEST), YOU WILL LOOP ON THE ILLEGAL FUNCTION.

20. TEST PAR IN RSER

SET PAR IN RSER AND CHECK. ALSO TEST ERR IN RSDS TO SET BECAUSE OF THE PAR SETTING.

21. TEST DPR AND MOL IN RSDS

BOTH THESE BITS SHOULD BE SET IN RSDS IF THE DRIVE IS ON LINE AND UP TO SPEED.

22. LOOK AHEAD TEST

FIRST CHECK TO SEE IF SECTOR FRACTION BITS ARE MOVING. NOW SET RSDA TO 0 AND INCREMENT IT EVERY TIME THE ADDR.CONF BIT SETS. IF THE ADDR.CONF BIT DOES NOT SET IN A CERTAIN LENGTH OF TIME, A TIME OUT ERROR OCCURS.

23. PARITY TEST

THIS WILL TEST THE PARITY LOGIC ONLY IF THERE IS PARITY MEMORY ON THE SYSTEM IN LESS THAN 28K. IT WILL WRITE BAD PARITY IN A MEMORY LOCATION THEN TRY TO DO A WRITE TO THE DRIVE FROM THAT LOCATION. THIS SHOULD CAUSE A PARITY ERROR.

24. TEST WRITE CHECK ERROR

IN THIS TEST THE PROGRAM WRITES A -1 ON TO THE DISK. A 0 IS NOW FLOATED THROUGH THE WORD IN THE BUS ADDRESS LOCATION, AND A WRITE CHECK FUNCTION IS DONE. THE WCE BIT IN RSCS2 SHOULD SET AND SHOULD CAUSE THE TRE BIT IN RSCS1 TO SET. THESE BITS

ARE THEN CLEARED. A WORD OF 0 IS NOW WRITTEN ON THE DISK AND  
A 1 IS FLOATED THROUGH THE WORD IN THE BUS ADDRESS AND THE

CZRSB-G RH11-RS03/LA-RS04 BASIC FUNCTION DIAGNOSTIC PAGE 10  
DESCRIPTION

WRITE CHECK FUNCTION TEST IS REPEATED.

25. TEST PROGRAM ERROR BIT IN RSCS2

HERE THE PROGRAM ATTEMPTS TO INITIATE A DATA TRANSFER OPERATION WHILE THE CONTROL IS CURRENTLY PERFORMING ONE. THIS SHOULD CAUSE PGE TO SET WHICH SHOULD CAUSE TRE TO SET. THESE BITS ARE THEN CLEARED. RSWC IS ALSO TESTED FOR IT SHOULD NOT BE 0 FOR THE CURRENT OPERATION SHOULD HAVE BEEN ABORTED DUE TO THE PGE ERROR.

26. TEST RMR IN RSER

HERE A WRITE COMMAND IS GIVEN AND DURING ITS EXECUTION THE PROGRAM TRYS TO MODIFY THE RSDA REG. THIS SHOULD CAUSE THE RMR BIT TO SET WHICH CAUSES THE ERR BIT TO SET. THESE BITS ARE THEN CLEARED.

27. TEST DCK IN RSER

HERE A WRITE COMMAND IS GIVEN THEN DURING THIS FUNCTION A DRIVE CLEAR COMMAND IS GIVEN. THIS SHOULD CAUSE THE DCK BIT TO SET WHICH SHOULD CAUSE THE ERR BIT TO SET. THESE BITS ARE THEN CLEARED.

28. TEST DISK ADDRESS REGISTER

LOAD THE LAST DISK ADDRESS (7777) INTO RSDA. DO A ONE WORD WRITE AND CHECK THAT RSDA INCREMENTED TO 10000.

29. TEST IAE ERROR

DO A ONE WORD WRITE BUT FIRST SET RSDA TO AN INVALID ADDRESS SUCH AS 10000. THIS SHOULD CAUSE A IAE ERROR WHICH WILL CAUSE ERR, ATA AND SC BITS TO SET. THESE BITS ARE THEN CLEARED BY LOADING A 1 INTO ATA IN RSAS.

30. TEST FOR NON-EXISTENT DISK ERROR

FIRST FIND A DRIVE THAT IS NOT ON THE SYSTEM OR OFF LINE. NOW TRY TO DO A ONE WORD WRITE TO THAT DRIVE. NED IN RSCS2 SHOULD SET WHICH SHOULD CAUSE TRE TO SET. THESE BITS ARE THEN CLEARED BY MOVING A 1 INTO TRE.

31. TEST DAO IN RSER AND LBT IN RSDS

SET RSDA TO ITS LAST ADDRESS. NOW WRITE ONE SECTOR PLUS ONE WORD. DAO SHOULD SET AND LBT SHOULD SET. THESE SHOULD CAUSE ERR, ATA, TRE AND SC TO SET. THESE ARE CLEARED BY DOING A CLEAR.

32. TEST BAI IN RSCS2



SET BAI IN RSCS2. DO A ONE WORD WRITE AND CHECK RSBA TO SEE



```

605 000176 000000          SWREG: 0
606
607          000200          . = 200
608 000200 000137 000230    JMP @#STRT
609
610          000220          . = 220
611 000220 000137 024444    JMP @#WRTLCK      ;WRITE LOCK TEST
612
613          000230          . = 230
614 000230 042737 004000 001160  STRT: BIC #BIT11,ONCEE ;CLEAR TKSEL TEST FLAG
615 000236 000137 001202    JMP @#BEGIN      ;STARTING ADDRESS IS BEGIN
616
617          ;STARTING ADDRESSES FOR SWITCH REGISTER TESTING OF RH-RS REGISTERS
618
619          000250          . = 250
620 000250 000137 023556    JMP @#SRSWC      ;WORD COUNT REG
621 000254 000137 023602    JMP @#SRSBA      ;BUS ADDR REG
622 000260 000137 023626    JMP @#SRSDA      ;DISK ADDR REG
623 000264 000137 023652    JMP @#SRSDS      ;DRIVE STATUS REG
624 000270 000137 023676    JMP @#SRSER      ;ERROR REG
625 000274 000137 023722    JMP @#SRSLA      ;LOOK AHEAD REG
626 000300 000137 023732    JMP @#SRCS2      ;CS2 REG
627 000304 000137 023756    JMP @#SRAS      ;ATTENTION SUMMARY REG
628 000310 000137 024002    JMP @#RSMRR      ;MAINTENANCE REG
629 000314 000137 024026    JMP @#SRSCS1     ;CS1 REG
630
631          000320          . = 320
632 000320 052737 004000 001160  BIS #BIT11,ONCEE ;SET TKSEL TEST FLAG
633 000326 000137 001202    JMP @#BEGIN      ;ENABLES OPERATOR TO DO A
634          ;WRITE OR READ TO A DESIRED ADDR
635          ;OPTIONAL WC, DA, AND UNIT #
636
  
```

```
637
638 000001
639 104000
640 177776
641 177776
642 000007
643 000000
644 000001
645 000002
646 000003
647 000004
648 000005
649 000006
650 000007
651 000001
652 000002
653 000004
654 000010
655 000020
656 000040
657 000100
658 000200
659 000400
660 001000
661 002000
662 004000
663 010000
664 020000
665 040000
666 100000
667 000001
668 000000
669

N= 1
HLT= EMT
PS= 177776
PSW= PS
BELL= 7
R0= %0
R1= %1
R2= %2
R3= %3
R4= %4
R5= %5
SP= %6
PC= %7
BIT0= 1
BIT1= 2
BIT2= 4
BIT3= 10
BIT4= 20
BIT5= 40
BIT6= 100
BIT7= 200
BIT8= 400
BIT9= 1000
BIT10= 2000
BIT11= 4000
BIT12= 10000
BIT13= 20000
BIT14= 40000
BIT15= 100000

GOOD= %1
BAD= %0

;INITALIZE FOR NEWTST
;SET HLT TO EMT FOR ERROR TYPEOUTS
;PROCESSOR STATUS
;PROCESSOR STATUS WORD
;BELL
;R0 - DEFINE REGISTERS
;R1
;R2
;R3
;R4
;R5
;R6 - STACK POINTER
;R7 - PROGRAM COUNTER
;BIT EQUATES
;FOR GOOD DATA
;FOR BAD DATA
```

```

670          001000          . =      1000
671
672 001000 000000          ICNT: 0          ;LH = ITERATION COUNT ;RH = TEST NO.
673 001002 000000          ERRORS: 0         ;ERROR COUNT
674 001004 000000 000000 PCNT: 0,0       ;2 WORD PASS COUNT
675 001010 000000          LAD: 0          ;LOOP ADDRESS FOR SCOPE
676 001012 000000          HLTADR: 0         ;ADDRESS OF LAST HLT INSTRUCTION EXECUTED
677 001014 001000          FILCHR: 1000       ;FILCHR=0 (CHAR) ;FILCHR+1=2 (COUNT)
678 001016 177564          TPS: 177564      ;OUTPUT STATUS REGISTER
679 001020 177560          TKS: 177560
680 001022 177562          TKB: 177562
681 001024 177566          TPB: 177566
682 001026 177570          SWR: 177570
683 001030 177570          DISPLAY:177570 ;OUTPUT BUFFER
684
685          001100          . =      1100
686 001100 000000          SAVBAD: 0       ;LOC FOR ILLEGAL FUNCTION CODE
687 001102 000000          OBUFSV: 0      ;LOC OF OUTBUF
688
689          ;DISK I/O REGISTERS
690
691 001104 172040          RSCS1: 172040   ;DISK CONTROL + STATUS REGISTER
692 001106 172050          RSCS2: 172050   ;DISK CONTROL + STATUS REGISTER
693 001110 172042          RSWC: 172042   ;WORD COUNT REGISTER
694 001112 172044          RSBA: 172044   ;BUS ADDRESS
695 001114 172046          RSDA: 172046   ;DISK ADDRESS (DESIRED ADDRESS)
696 001116 172052          RSDS: 172052   ;DRIVE STATUS
697 001120 172054          RSER: 172054   ;ERROR REG.
698 001122 172056          RSAS: 172056   ;ATTENTION SUMMARY
699 001124 172060          RSLA: 172060   ;LOOK AHEAD
700 001126 172062          RSDB: 172062   ;DATA BUFFER REGISTER
701 001130 172064          RSMR: 172064   ;MAINTENANCE REGISTER
702 001132 172066          RSDT: 172066   ;DRIVE TYPE REGISTER
703 001134 000204          RSVEC: 204     ;INTERUPT VECTOR
704 001136 000206          RSVCP: 206     ;INTERUPT PRIO. VECTOR
705 001140 172041          RSCS1B: 172041 ;ODD BYTE ADD FOR CS1
706 001142 172051          RSCS2B: 172051 ;ODD BYTE ADD FOR CS2
707 001144 172043          RSWCB: 172043  ;ODD BYTE ADD FOR CW
708 001146 172045          RSBAB: 172045  ;ODD BYTE ADD FOR BA
709
710          ;MEMORY MANAGEMENT REGISTER ASSIGNMENTS
711
712          177572          SR0=177572
713          172340          KIPAR0=172340
714          172342          KIPAR1=172342
715          172344          KIPAR2=172344
716          172356          KIPAR7=172356
717          172300          KIPDR0=172300
718          172302          KIPDR1=172302
719          172304          KIPDR2=172304
720          172316          KIPDR7=172316
721          000006          RW=6
722          000000          UP=00
723
    
```

```

724 ;BIT ASSIGNMENTS FOR ERROR TYPEOUTS
725 ;THE RS REGISTERS ARE DIVIDED INTO 3 GROUPS.
726 ;CS1,CS2 AND ER ARE IN THE FIRST GROUP.THIS GROUP IS ALWAYS
727 ;TYPED WITH EITHER OF THE OTHER GROUPS. AS,BA,DA, WC AND DS
728 ;ARE IN THE SECOND GROUP. DT,DB,MR, AND LA ARE IN THE 3RD
729 ;GROUP.YOU CAN NOT INTERMIX GROUP 2 OR 3. THEY HAVE
730 ;TO BE TYPED SEPERATELY.
731 ;EXAMPLE: HLT !CS1,AS,BA
732 ;          HLT !CS1!DT!DB
733
  
```

734	000001	CS1=1	:CONTROL AND STATUS 1
735	000002	ER=2	:CONTROL AND STATUS 2
736	000004	DA=4	:DESIRED ADD
737	000010	WC=10	:WORD COUNT
738	000020	BA=20	:BUS ADDRESS
739	000040	DS=40	:DRIVE STATUS
740	000100	AS=100	:ATTENTION SUMMARY
741	000200	CS2=200	:CONTROL AND STATUS REG
742	000204	LA=204	:LOOK AHEAD
743	000210	DB=210	:DATA BUFFER
744	000220	MR=220	:MAINTENANCE
745	000240	DT=240	:DRIVE TYPE

:BIT ASSIGNMENTS FOR THE REGISTER BITS

749	040000	TRE=40000	:TRANSFER ERROR CS1
750	100000	SC=100000	:SPECIAL CONDITIONS CS1
751	000100	IR=100	:INPUT READY CS2
752	000200	OR=200	:OUTPUT READY CS2
753	002000	PGE=2000	:PROGRAM ERROR-CS2
754	010000	NED=10000	:NON-EXISTENT DRIVE CS2
755	040000	WCE=40000	:WRITE CHECK ERROR-CS2
756	100000	DLT=100000	:DATA LATE ERROR CS2
757	000200	DRY=200	:DRIVE READY DS
758	020000	PIP=20000	:POSITIONING IN PROGRESS DS
759	002000	LBT=2000	:LAST BLOCK TRANSFER-DS
760	040000	ERR=40000	:ERROR DS
761	100000	ATA=100000	:ATTENTION ACTIVE-DS
762	001000	DAO=1000	:DISK OVERFLOW ERROR-ER
763	100000	DCK=100000	:DATA CHECK ERROR-ER
764	000010	BAI=10	:BUS ADDP. INCREMENT INHIBIT
765	000100	IE=100	:INTERRUPT INABLE CS1

```

766 ;WORKING LOCATIONS
767
768 001150 146723 RANNU: 146723 ;RANDOM NUMBER PRIME
769 001152 000000 UNNUM: 0 ;UNIT CURRENTLY BEING TESTED
770 001154 000000 UNITSV: 0 ;SET BIT=UNIT ON BUS
771 001156 000000 UNCMP: 0 ;FOR COMPARING FOR # OF DEVICE
772 001160 000000 ONCEE: 0 ;DID WE TEST ANY DRIVES
773 001162 000000 RS04DT: 0 ;CLR IF RS03 SET IF RS04
774 001164 000000 TIMSV: 0 ;SAVE LOC FOR TIME
775 001166 000000 AOB1: 0 ;PORT SWITCH
776 000004 WWP=4 ;WRITE WRONG PARITY
777 172100 MPRO=172100 ;PARITY REG
778 001170 000000 BPORTT: 0 ;BUFFER ADDR FOR -B- PORT
779 001172 000000 SAVEE: 0 ;WORK LOC
780
781 ;DISCRIPTION OF ONCEE BITS
782 ;BIT0 MEANS FOUND DRIVE
783 ;BIT11 DO TKSEL TEST
784 ;BIT12 TYPE COULD NOT FIND NED ONLY ONCE
785 ;BIT13 TYPE NO MEM ON B PORT ONLY ONCE
786 ;BIT14 0- DO WCE WITH 0 -1 DO WCE WITH 1
787 ;BIT15 MEANS ERROR FOUND
788
789 ;RH11 WORK REGISTERS
790 ;(CAN BE CHANGED IN ANY ROUTINE)
791 001174 000000 WORK: 0
792 001176 000000 WORK1: 0
793 001200 000000 WORK2: 0
794
795 001202 012706 000500 BEGIN: MOV #500,SP ;SET STACK TO *** 500 ***
796 001206 012737 023060 000024 MOV #.POWER,@#24 ;SET UP PF VECTOR
797 001214 012737 000340 000026 MOV #340,@#26 ;LOCK OUT THE WORLD
798 001222 012737 022506 000030 MOV #.HLT,@#30 ;SET EMT VECTOR
799 001230 012737 000340 000032 MOV #340,@#32 ;LOCK UP
800 001236 012737 023512 000034 MOV #.TRAP,@#34 ;SET TRAP VECTOR
801 001244 012737 000340 000036 MOV #340,@#36 ;LOCK UP
802 001252 005037 001000 CLR ICNT ;INIT ICNT
803 001256 005037 001010 CLR LAD ;INIT LAD
804 001262 042737 143777 001160 BIC #143777,ONCEE ;CLEAR ONCEE
805 001270 012737 027454 001102 MOV #OUTBUF,OBUFSV ;SAVE LOC OF OUTBUFFER
806 001276 032737 004000 001160 BIT #BIT11,ONCEE ;DO TKSEL TEST?
807 001304 001402 BEQ .+6 ;NO
808 001306 000137 024104 JMP @#TKSEL ;YES
809 001312 104420 SUSWR ;SIZE FOR HDWR SWR
810 001314 005227 177777 INC #-1 ; SO TITLE PRINTOUT 1ST PASS ONLY
811 001320 001032 BNE AROUND ; THRU FOR PASSES >1
812 ;*****
813 ; ROUTINE TO PRINT DIAGNOSTICS TITLE
814 ;*****
815 001322 104402 001326 TYPE ..+2 ;.ASCIZ 'CZRSB-G RH11-RS03,RS03LA,RS04 DIAGNOSTIC '*
816 001406 000240 AROUND: NOP ; FALL THRU
817

```

```

818 ;NOW TEST FOR DRIVES
819 ;*****
820 ;TEST 1 TEST FOR DRIVES
821 ;*****
822 001410 104400 TST1: SCOPE
823
824 001412 012701 000010 MULTII: MOV #8,R1 ;PUT 8 INTO R1 FOR COUNT
825 001416 005077 177464 CLR @RSCS2 ;SET DEVICE TO ZERO
826 001422 012777 000007 177470 TRY: MOV #7,@RSER ;CAUSE AN ERROR +SETS BIT IN RSAS REG
827 001430 005301 DEC R1 ;DO A MAXIMUM OF 8 TIMES
828 001432 001403 BEQ DVNUM ;TESTED FOR ALL DRIVES GET OUT
829 001434 005277 177446 INC @RSCS2 ;INCREMENT DRIVE UNIT
830 001440 000770 BR TRY ;REPEAT FOR NEXT DRIVE
831 001442 017737 177454 001154 DVNUM: MOV @RSAS,UNITSV ;SAVE
832 001450 012737 000401 001156 MOV #401,UNCMP ;SETUP TO CMP WITH UNITSV
833 001456 012737 000000 001152 MOV #0,UNNUM ;PUT 0 INTO UNIT NO.
834 001464 032777 020000 177334 BIT #BIT13,@SWR ;INHIBIT TYPE OUT?
835 001472 001015 BNE SITEST ;YES
836 001474 104402 001500 TYPE ..+2 ;.ASCIZ <15><12>'TESTING UNIT ''
837 001520 042737 100000 001160 BIC #BIT15,ONCEE ;CLEAR ERROR FLAG
838 001526 033737 001156 001154 SITEST: BIT UNCMP,UNITSV ;IS THIS DRIVE ON THE SYSTEM
839 001534 001475 BEQ TRYNX ;NO
840 001536 013777 001152 177342 MOV UNNUM,@RSCS2 ;YES PUT UNIT # INTO CS2
841 001544 022777 000004 177360 CMP #4,@RSDT ;IS THIS A RS03LA?
842 001552 001004 BNE 7$ ;NO
843 001554 012737 000004 001162 MOV #4,RS04DT ;SETUP DRIVE TYPE FOR A LA DISK
844 001562 000435 BR 1$ ;GET OUT
845 001564 022777 000000 177340 7$: CMP #0,@RSDT ;IS THIS A RS03?
846 001572 001003 BNE 2$ ;NO
847 001574 005037 001162 CLR RS04DT ;YES
848 001600 000426 BR 1$ ;GET OUT
849 001602 022777 000001 177322 2$: CMP #1,@RSDT ;IS THIS A RS03 4US?
850 001610 001003 BNE 3$ ;NO
851 001612 005037 001162 CLR RS04DT ;RS03
852 001616 000417 BR 1$ ;GET OUT
853 001620 022777 000002 177304 3$: CMP #2,@RSDT ;IS THIS A RS04?
854 001626 001410 BEQ 6$ ;YES
855 001630 022777 000003 177274 CMP #3,@RSDT ;IS IT A RS04?
856 001636 001404 BEQ 6$ ;YES
857 001640 043737 001156 001154 BIC UNCMP,UNITSV ;CLEAR UNWANTED ATA BIT
858 001646 000430 BR TRYNX ;GET A NEW NUMBER
859 001650 052737 177777 001162 6$: BIS #-1,RS04DT ;YES RS04
860 001656 032777 020000 177142 1$: BIT #BIT13,@SWR ;INHIBIT TYPE OUT?
861 001664 001020 BNE 4$ ;YES
862 001666 032737 100000 001160 BIT #BIT15,ONCEE ;ANY ERRORS?
863 001674 001404 BEQ 5$ ;NO
864 001676 104402 001702 TYPE ..+2 ;.ASCIZ <15><12><12>
865 ;*****4
866 001706 5$:
867 001706 013746 001152 MOV UNNUM,-(6) ;PUT UNNUM ON STACK
868 001712 104406 TYPES ;TYPE STACK IN OCTAL - SUPRESS
869 ;*****
870 001714 104402 000040 TYPE ,40 ;TYPE SPACE
871 001720 042737 100000 001160 BIC #BIT15,ONCEE ;CLEAR ERROR FLAG
872 001726 000473 4$: BR NC:GO ;NOW TEST
    
```



```

873 001730 006337 001156      TRYNX: ASL      UNCOMP      ;CHECK NEXT BIT FOR DRIVE
874 001734 103403              BCS      CHCKDV      ;DID WE TEST ANY REG?
875 001736 005237 001152      INC      UNNUM       ;INC UNIT #
876 001742 000671              BR       STTEST      ;CHECK FOR NEXT DRIVE
877
878 001744 032737 000001 001160  CHCKDV: BIT      #BIT0,ONCEE  ;DID WE TEST ANY DRIVES?
879 001752 001057              BNE      DONEE      ;YES WE DID TEST A DRIVE
880 001754 012737 100000 001156  MOV      #100000,UNCOMP ;NO DRIVES TESTED, COULD NOT SET
881 001762 005037 001152      CLR      UNNUM       ;ANY AS BITS, THUS DEFAULTS TO
882 001766 032777 020000 177032  BIT      #BIT13,@SWR  ;INHIBIT TYPE OUT?
883 001774 001045              BNE      4$         ;YES
884 001776 013746 001152      MOV      UNNUM,-(6)  ;PUT UNNUM ON STACK
885 002002 104406              TYPES     ;TYPE STACK IN OCTAL - SUPRESS
886 002004 104402 000040      TYPE     ,40        ;TYPE SPACE
887 002010 104402 002014      TYPE     ,.+2       ;.ASCIZ <15><12>'COULD NOT FIND DRIVE WILL TEST DRIVE 0
888 002106 000000              HALT
889 002110 000402      4$: BR       NOWGO   ;TEST DRIVE 0
890 002112 000137 022052  DONEE: JMP      DONE   ;GET OUT
891
892      ;THIS TEST IS DESIGNED TO TEST THE ABILITY OF RESET
893      ;TO CLEAR ALL THE RH AND RS REGISTERS
894
895 002116 012737 027454 001102  NOWGO: MOV      #OUTBUF,OBUSV ;SAVE LOC OF OUTBUFFER
896 002124 052737 000001 001160  BIS      #BIT0,ONCEE  ;SET FOUND DRIVE FLAG
897 002132 013737 022504 001164  MOV      TIMES,TIMSV ;SAVE TIME
898 002140 012737 000001 022504  MOV      #1,TIMES    ;ONLY TEST ONCE
899
900      ;*****
901      ;TEST 2      RESET TEST FOR REGISTERS
902      ;*****
903 002146 104400      TST2: SCOPE
904 002150 012737 000340 177776  MOV      #340,@#PS   ;LOCK OUT INTERUPTS
905 002156 013777 001152 176722  MOV      UNNUM,@RSCS2 ;GET UNIT #
906 002164 012777 177776 176712  MOV      #177776,@RSCS1 ;SET ALL
907 002172 012777 177777 176712  MOV      #177777,@RSBA ;POSSIBLE R/W
908 002200 012777 177777 176706  MOV      #177777,@RSDA ;BITS IN THESE REGISTERS
909 002206 012777 177777 176704  MOV      #177777,@RSER
910 002214 012777 177777 176706  MOV      #177777,@RSMR
911 002222 012777 177777 176660  MOV      #177777,@RSWC
912 002230 012777 177737 176650  MOV      #177737,@RSCS2
913 002236 000005      RESET      ;CLEAR ALL BITS IN ALL REG.
914
915      ;TEST RSCS2 FOR CLEARED BITS
916 002240 022777 000100 176640  CMP      #100,@RSCS2 ;DID THESE BITS GET CLEARED?
917 002246 001401              BEQ      ,+4        ;YES
918 002250 104200              HLT      !CS2      ;(417) SHOULD BE CLEARED IN CS2
919 002252 013777 001152 176626  MOV      UNNUM,@RSCS2 ;PUT # OF UNIT IN TEST IN CS2
920 002260 022777 010600 176630  CMP      #10600,@RSDS ;IS DPR AND MOL SET?
921 002266 001401              BEQ      ,+4        ;YES
922 002270 104040              HLT      !DS      ;NO WHY NOT?
923
924 002272 022777 004200 176604  ;TEST CONTROL AND STATUS REG 1
925 002300 001401              CMP      #4200,@RSCS1 ;DID THE READY BIT SET?
926 002302 104001              BFQ      ,+4        ;YES
                      HLT      !CS1      ;READY SHOULD BE SET
    
```

```

927 ;TEST BUS ADDRESS REGISTER
928
929 002304 005777 176602 TST @RSBA ;IS BA REG. CLEARED
930 002310 001401 BEQ .+4 ;YES
931 002312 104020 HLT !BA ;SHOULD BE 0
932
933 ;TEST DISK ADDRESS REGISTER
934
935 002314 005777 176574 TST @RSDA ;IS DA CLEARED
936 002320 001401 BEQ .+4 ;YES
937 002322 104004 HLT !DA ;SHOULD BE 0
938
939 ;TEST ERROR REG RSER
940
941 002324 005777 176570 TST @RSER ;DID RSER CLEAR?
942 002330 001401 BEQ .+4 ;YES
943 002332 104002 HLT !ER ;BITS(157015) SHOULD BE CLEARED
944
945 ;TEST RS MAINTENANCE REGISTER
946
947 002334 032777 000077 176566 BIT #77,@RSMR ;DID THESE BITS GET CLEARED
948 002342 001401 BEQ .+4 ;YES
949 002344 104220 HLT !MR ;BITS(77) SHOULD BE 0
950
951 ;TEST WC REG IT SHOULD NOT CHANGE
952
953 002346 022777 177777 176534 CMP #177777,@RSWC ;DID IT CHANGE?
954 002354 001401 BEQ .+4 ;NO
955 002356 104010 HLT !WC ;RESET SHOULD NOT MODIFY RSWC
956
957 ;TEST RSAS
958
959 002360 005777 176536 TST @RSAS ;IS REG CLEAR
960 002364 001401 BEQ .+4 ;YES
961 002366 104100 HLT !AS ;NO
  
```

```

962 :*****
963 :TEST 3 TEST CLEAR BIT IN CS2 ON ALL THE R/W BITS
964 :*****
965 002370 104400 TST3: SCOPE
966
967 002372 012737 000340 177776 TTAGG: MOV #340,@#PS ;LOCK OUT INTERRUPTS
968 002400 013777 001152 176500 MOV UNNUM,@RSCS2 ;GET UNIT #
969 002406 012777 043576 176470 MOV #43576,@RSCS1 ;SET ALL
970 002414 012777 177777 176470 MOV #177777,@RSBA ;POSSIBLE
971 002422 012777 177777 176464 MOV #177777,@RSDA ;REGISTERS
972 002430 012777 177777 176462 MOV #177777,@RSER
973 002436 012777 177777 176462 MOV #177777,@RSDB
974 002444 012777 177777 176436 MOV #177777,@RSWC
975 002452 012777 020417 176426 MOV #20417,@RSCS2
976 002460 012777 000071 176442 MOV #71,@RSMR
977 002466 012777 000040 176412 MOV #40,@RSCS2 ;CLEAR ALL BITS
978 002474 022777 000100 176404 CMP #100,@RSCS2 ;DID THE RIGHT BITS CLEAR?
979 002502 001401 BEQ .+4 ;YES
980 002504 104200 HLT !CS2 ;(417) SHOULD BE CLEARED IN CS2
981 002506 013777 001152 176372 MOV UNNUM,@RSCS2 ;GET DRIVE NUMBER
982 002514 032777 173577 176362 BIT #173577,@RSCS1 ;DID ALL BITS GET CLEARED
983 002522 001401 BEQ .+4 ;YES
984 002524 104001 HLT !CS1 ;NO, ALL BITS SHOULD BE 0
985
986 ;TEST BUS ADDRESS REGISTER
987 002526 005777 176360 TST @RSBA ;IS BA REG. CLEARED
988 002532 001401 BEQ .+4 ;YES
989 002534 104020 HLT !BA ;SHOULD BE 0
990
991 ;TEST DISK ADDRESS REGISTER
992
993 002536 005777 176352 TST @RSDA ;IS DA CLEARED
994 002542 001401 BEQ .+4 ;YES
995 002544 104020 HLT !BA ;SHOULD BE 0
996
997 ;TEST ERROR REG RSER
998
999 002546 005777 176346 TST @RSER ;DID THESE BITS GET CLEARED
1000 002552 001401 BEQ .+4 ;YES
1001 002554 104002 HLT !ER ;BITS(157015) SHOULD BE CLEARED
1002
1003 ;TEST RS MAINTENANCE REGISTER
1004 002556 032777 000077 176344 BIT #77,@RSMR ;DID THESE BITS GET CLEARED
1005 002564 001401 BEQ .+4 ;YES
1006 002566 104220 HLT !MR ;BITS(77) SHOULD BE 0
1007
1008 ;TEST WC REG. IT SHOULD NOT CHANGE
1009 002570 022777 177777 176312 CMP #177777,@RSWC ;DID WC CHANGE
1010 002576 001401 BEQ .+4 ;NO
1011 002600 104010 HLT !WC ;WHY DID IT CHANGE?
  
```

```

1012 :*****
1013 :TEST 4          SET AND CLEAR ALL REGISTERS
1014 :*****
1015 002602 104400 TST4: SCOPE
1016 :CAN WE SET THE FUNCTION BITS IN THE RSCS1 REG.
1017 :BITS 7,6,5,4,3,2&1
1018
1019 002604 104414          CLRDK          :CLEAR ALL RS REG
1020 002606 013737 001164 022504      MOV          TIMSV,TIMES      :GET TIME
1021 002614 012777 003576 176262      MOV          #3576,@RSCS1    :SET DISK FUNCTION BITS
1022 002622 022777 007776 176254      CMP          #7776,@RSCS1    :ARE THESE BITS SET?
1023 002630 001401          BEQ          .+4              :NO
1024 002632 104001          HLT          !CS1            :SHOULD = 7776
1025 002634 012777 002524 176242      MOV          #2524,@RSCS1    :SET THESE BITS
1026 002642 022777 006724 176234      CMP          #6724,@RSCS1    :DID THEY SET
1027 002650 001401          BEQ          .+4              :YES
1028 002652 104001          HLT          !CS1            :SHOULD BE 6724
1029 002654 012777 001052 176222      MOV          #1052,@RSCS1    :SET THESE BITS
1030 002662 022777 005252 176214      CMP          #5252,@RSCS1    :ARE THEY =?
1031 002670 001401          BEQ          .+4              :YES
1032 002672 104001          HLT          !CS1            :SHOULD = 5252
1033 002674 104400 TST5: SCOPE
1034 :CLEAR THE FUNCTION BITS
1035
1036 002676 012777 043576 176200      MOV          #43576,@RSCS1   !SET DISK FUNCTION BITS
1037 002704 005077 176174          CLR          @RSCS1
1038 002710 022777 004200 176166      CMP          #4200,@RSCS1    :IS THE READY BIT SET
1039 002716 001401          BEQ          .+4              :YES
1040 002720 104001          HLT          !CS1            :RSCS1 SHOULD = 4200
1041
1042 :*****
1043 :TEST 6          TEST RSCS2
1044 :*****
1045 002722 104400 TST6: SCOPE
1046
1047 002724 000005          RESET         :CLEAR WORLD
1048 002726 022777 000100 176152      CMP          #100,@RSCS2     :DID THEY CLEAR?
1049 002734 001401          BEQ          .+4              :YES
1050 002736 104200          HLT          !CS2            :NO
1051 002740 012777 021037 176140      MOV          #21037,@RSCS2   :SET BITS 21017
1052 002746 022777 021137 176132      CMP          #21137,@RSCS2   :DID THESE BITS GET SET
1053 002754 001405          BEQ          1$              :YES
1054 002756 017700 176124          MOV          @RSCS2,BAD
1055 002762 012701 021137          MOV          #21137,GOOD
1056 002766 104000          HLT
:WHAT CS2 SHOULD =
:CS2 = BAD GOOD = CORRECT ANS
    
```

```

1057 002770 012777 020025 176110 1$: MOV #20025,@RSCS2 ;SET THESE BITS
1058 002776 022777 020125 176102 CMP #20125,@RSCS2 ;DID THESE BITS GET SET
1059 003004 001401 BEQ .+4 ;YES
1060 003006 104200 HLT !CS2 ;NO,CS2 SHOULD = 20125
1061 003010 012777 000012 176070 MOV #12,@RSCS2 ;LOAD THESE BITS
1062 003016 022777 000112 176062 CMP #112,@RSCS2 ;DID THESE BITS GET SET IN CS2
1063 003024 001401 BEQ .+4 ;YES
1064 003026 104200 HLT !CS2 ;BAD = CS2 GOOD = CORRECT ANS
1065 003030 012777 177777 176050 MOV #-1,@RSCS2 ;SET BITS
1066 003036 005077 176044 CLR @RSCS2 ;CLEAR THEM
1067 003042 022777 000100 176036 CMP #100,@RSCS2 ;DID CLEAR WORK
1068 003050 001401 BEQ .+4 ;YES
1069 003052 104200 HLT !CS2 ;R/W BITS DID NOT CLEAR
1070 003054 013777 001152 176024 MOV UNNUM,@RSCS2 ;GET UNIT #
1071 003062 104400 TST7: SCOPE
1072 ;CAN WE SET ALL THE RSBA BITS
1073
1074 003064 012777 177777 176020 MOV #177777,@RSBA ;SET THE BITS
1075 003072 022777 177776 176012 CMP #177776,@RSBA ;DID THEY SET
1076 003100 001401 BEQ .+4 ;YES
1077 003102 104020 HLT !BA ;BITS 17776 SHOULD BE SET
1078 003104 012777 125252 176000 MOV #125252,@RSBA ;SET THESE BITS
1079 003112 022777 125252 175772 CMP #125252,@RSBA ;ARE THEY =
1080 003120 001401 BEQ .+4 ;YES
1081 003122 104020 HLT !BA ;SHOULD BE 125252
1082 003124 012777 052524 175760 MOV #52524,@RSBA ;SET THESE BITS
1083 003132 022777 052524 175752 CMP #52524,@RSBA ;ARE THEY =
1084 003140 001401 BEQ .+4 ;YES
1085 003142 104020 HLT !BA ;SHOULD BE 52524
1086
1087 003144 104400 TST10: SCOPE
1088 ;FLOAT A 1 THROUGH RSBA
1089
1090 003146 012701 000002 FLOTBA: MOV #2,GOOD ;GET A 2
1091 003152 000241 CLC ;CLEAR CARRY
1092 003154 010177 175732 1$: MOV GOOD,@RSBA ;FLOAT NUMBER
1093 003160 017700 175726 MOV @RSBA,BAD ;GET BA
1094 003164 020100 CMP GOOD,BAD ;COMPARE BA
1095 003166 001401 BEQ .+4 ;BA CORRECT
1096 003170 104000 HLT ;BAD=BA GOOD=CORRECT ANS
1097 003172 006101 ROL GOOD ;ROTATE NUMBER
1098 003174 103367 BCC 1$ ;LOOP TILL DONE
    
```

```

1099 003176 104400 TST11: SCOPE
1100
1101 ;CLEAR THE RSBA REGISTER
1102
1103 003200 012777 177777 175704 MOV #177777,@RSBA ;SET RSBA EQUAL TO ALL ONES
1104 003206 005077 175700 CLR @RSBA
1105 003212 005777 175674 TST @RSBA ;TEST FOR BIT0 SET IN RSBA (READ ONLY BIT)
1106 003216 001401 BEQ .+4 ;YES
1107 003220 104020 HLT !BA ;NO
1108 003222 104400 TST12: SCOPE
1109
1110 ;CAN WE SET ALL BITS IN RSWC REGISTER
1111
1112 003224 012777 177777 175656 MOV #177777,@RSWC ;SET WC BITS
1113 003232 022777 177777 175650 CMP #177777,@RSWC ;ARE ALL BITS SET
1114 003240 001401 BEQ .+4 ;YES
1115 003242 104010 HLT !WC ;NO
1116 003244 012777 125252 175636 MOV #125252,@RSWC ;SET THESE BITS
1117 003252 022777 125252 175630 CMP #125252,@RSWC ;ARE THEY =
1118 003260 001401 BEQ .+4 ;YES
1119 003262 104010 HLT !WC ;SHOULD BE 125252
1120 003264 012777 052525 175616 MOV #52525,@RSWC ;SET THESE BITS
1121 003272 022777 052525 175610 CMP #52525,@RSWC ;ARE THEY =
1122 003300 001401 BEQ .+4 ;YES
1123 003302 104010 HLT !WC ;SHOULD BE 152525
1124 003304 104400 TST13: SCOPE
1125
1126 ;FLOAT A 1 THROUGH RSWC
1127
1128 003306 012701 000001 FLOTWC: MOV #1,GOOD ;GET A 1
1129 003312 000241 CLC ;CLEAR CARRY
1130 003314 010177 175570 1$: MOV GOOD,@RSWC ;FLOAT NUMBER
1131 003320 017700 175564 MOV @RSWC,BAD ;GET WC
1132 003324 020100 CMP GOOD,BAD ;COMPARE WC
1133 003326 001401 BEQ .+4 ;WC CORRECT
1134 003330 104000 HLT ;BAD=WC GOOD=CORRECT ANS
1135 003332 006101 ROL GOOD ;ROTATE NUMBER
1136 003334 103367 BCC 1$ ;LOOP TILL DONE
  
```



```

1174 ;CAN WE CLEAR THE RSDA REG.
1175 003476 104400 TST17: SCOPE
1176
1177 003500 012777 177777 175406 MOV #177777,@RSDA ;SET RSDA TO ALL ONES
1178 003506 005077 175402 CLR @RSDA ;DID THEY SET
1179 003512 005777 175376 TST @RSDA ;TEST FOR ZERO RSDA
1180 003516 001401 BEQ .+4 ;YES
1181 003520 104004 HLT !DA ;ANS SHOULD BE 0
1182 003522 104400 TST20: SCOPE
1183
1184 ;SET AND CLEAR THE RSER REG.
1185
1186 003524 012777 177777 175366 MOV #177777,@RSER ;SET THESE BITS
1187 003532 022777 177017 175360 CMP #177017,@RSER ;DID THEY SET
1188 003540 001401 BEQ .+4 ;YES
1189 003542 104002 HLT !ER ;RSER SHOULD = 157017
1190 003544 112777 000001 175346 MOVB #1,@RSER ;A MOVB INST
1191 003552 022777 000001 175340 CMP #1,@RSER ;SHOULD MODIFY COMPLETE WD
1192 003560 001401 BEQ .+4 ;OK
1193 003562 104002 HLT !ER
1194
1195 003564 104400 TST21: SCOPE
1196
1197 003566 012777 052005 175324 MOV #52005,@RSER ;SET THESE BITS
1198 003574 022777 052005 175316 CMP #52005,@RSER ;DID THEY SET
1199 003602 001401 BEQ .+4 ;YES
1200 003604 104002 HLT !ER ;ER SHOULD = 52005
1201 003606 104400 TST22: SCOPE
1202
1203 003610 012777 125012 175302 MOV #125012,@RSER ;SET THESE BITS
1204 003616 022777 125012 175274 CMP #125012,@RSER ;DID THEY SET
1205 003624 001401 BEQ .+4 ;YES
1206 003626 104002 HLT !ER ;ER SHOULD = 105012
    
```



```

1207 003630 104400          TST23: SCOPE
1208
1209 003632 012777 177017 175260      MOV    #177017,@RSER    ;SET THESE BITS
1210 003640 005077 175254              CLR    @RSER           ;CLEAR THEM
1211 003644 005777 175250              TST    @RSER           ;DID THEY CLEAR
1212 003650 001401              BEQ    .+4             ;YES
1213 003652 104002              HLT    !ER            ;SHOULD = 0
1214 003654 104400          TST24: SCOPE
1215
1216                          ;SET AND CLEAR RSMR
1217
1218 003656 012777 000070 175244      MOV    #70,@RSMR      ;SET THESE BITS
1219 003664 017737 175240 001174      MOV    @RSMR,WORK     ;PUT INTO WORKABLE REG
1220 003672 042737 177700 001174      BIC    #177700,WORK   ;CLEAR JUNK
1221 003700 022737 000070 001174      CMP    #70,WORK       ;DID THEY SET
1222 003706 001401              BEQ    .+4             ;YES
1223 003710 104220              HLT    !MR            ;SHOULD = 70
1224 003712 104400          TST25: SCOPE
1225
1226 003714 012777 000070 175206      MOV    #70,@RSMR      ;SET BITS
1227 003722 005077 175202              CLR    @RSMR          ;CLEAR THEM
1228 003726 032777 000077 175174      BIT    #77,@RSMR      ;DID THEY CLEAR
1229 003734 001401              BEQ    .+4             ;YES
1230 003736 104220              HLT    !MR            ;BITS (77) SHOULD = 0
1231 003740 104400          TST26: SCOPE
1232
1233 003742 012777 000050 175160      MOV    #50,@RSMR      ;SET BITS
1234 003750 017737 175154 001174      MOV    @RSMR,WORK     ;PUT IN WORKABLE REG
1235 003756 042737 177700 001174      BIC    #177700,WORK   ;CLEAR JUNK
1236 003764 022737 000050 001174      CMP    #50,WORK       ;DID THESE BITS SET
1237 003772 001401              BEQ    .+4             ;YES
1238 003774 104220              HLT    !MR            ;BITS (50) SHOULD BE SET
1239 003776 104400          TST27: SCOPE
1240
1241 004000 012777 000020 175122      MOV    #20,@RSMR      ;SET BITS
1242 004006 017737 175116 001174      MOV    @RSMR,WORK     ;PUT INTO WORKABLE REG
1243 004014 042737 177700 001174      BIC    #177700,WORK   ;CLEAR JUNK
1244 004022 022737 000020 001174      CMP    #20,WORK       ;DID THEY SET
1245 004030 001401              BEQ    .+4             ;YES
1246 004032 104220              HLT    !MR            ;MR SHOULD AT LEAST HAVE A (21)
    
```

```

1247
1248
1249
1250 004034 104400
1251
1252 004036 012737 001000 001174 RANTS: MOV #1000,WORK ;MAKE TABLE 1000 WDS LONG
1253 004044 012701 027454 MOV #OUTBUF,R1 ;GET STARTING LOC OF TABLE
1254 004050 004537 026724 JSR R5,RANDOM ;GENERATE #
1255 004054 012704 027454 MOV #OUTBUF,R4 ;SETUP FOR COMPARE
1256 004060 012737 004066 001010 MOV #LOP1,LAD ;SETUP LOOP ADDR
1257 004066 012703 001000 LOP1: MOV #1000,R3 ;LOAD TEST COUNTER
1258 004072 005303 4$: DEC R3 ;DONE WITH COMPARE?
1259 004074 001413 BEQ 1$ ;YES
1260 004076 013705 001110 MOV RSWC,R5 ;GET WC ADDRESS
1261 004102 011415 MOV (R4),(R5) ;LOAD WC
1262 004104 021524 CMP (R5),(R4)+ ;IS IT CORRECT?
1263 004106 001771 BEQ 4$ ;YES
1264 004110 017700 174774 MOV @RSWC,BAD ;GET BAD WC
1265 004114 014401 MOV -(R4),GOOD ;GET GOOD ANS
1266 004116 104000 HLT ;TYPE THEM OUT
1267 004120 005724 TST (R4)+ ;UPDATE RANDOM NUMBER
1268 004122 000763 BR 4$ ;CONT
1269 004124 012704 027454 1$: MOV #OUTBUF,R4 ;GET STARTING LOC OF TABLE
1270 004130 012737 004136 001010 MOV #LOP2,LAD ;SETUP LOOP ADDR
1271 004136 012703 001000 LOP2: MOV #1000,R3 ;SETUP TEST COUNTER
1272 004142 005303 3$: DEC R3 ;DONE YET?
1273 004144 001413 BEQ 1$ ;YES
1274 004146 013705 001114 MOV RSDA,R5 ;LOAD DA ADDRESS INTO R5
1275 004152 011415 MOV (R4),(R5) ;LOAD DA
1276 004154 021524 CMP (R5),(R4)+ ;IS IT CORRECT?
1277 004156 001771 BEQ 3$ ;YES
1278 004160 017700 174730 MOV @RSDA,BAD ;GET BAD DATA
1279 004164 014401 MOV -(R4),GOOD ;GET GOOD DATA
1280 004166 104000 HLT ;TYPE IT OUT
1281 004170 005724 TST (R4)+ ;UPDATE RANDOM NUMBER
1282 004172 000763 BR 3$ ;CONTINUE
1283 004174 012704 027454 1$: MOV #OUTBUF,R4 ;GET STARTING LOC OF TABLE
1284 004200 012737 004206 001010 MOV #LOP3,LAD ;SETUP LOOP ADDR
1285 004206 012703 001000 LOP3: MOV #1000,R3 ;SETUP TEST COUNTER
1286 004212 005303 3$: DEC R3 ;DONE YET?
1287 004214 001416 BEQ 2$ ;YES
1288 004216 013705 001112 MOV RSBA,R5 ;LOAD ADDRESS OF BA INTO R5
1289 004222 011415 MOV (R4),(R5) ;LOAD BA
1290 004224 042714 000001 BIC #BIT0,(R4) ;CLEAR BIT 0
1291 004230 021514 CMP (R5),(R4) ;IS IT CORRECT?
1292 004232 001767 BEQ 3$ ;YES
1293 004234 017700 174652 MOV @RSBA,BAD ;GET BAD DATA
1294 004240 011401 MOV (R4),GOOD ;GET GOOD DATA
1295 004242 104000 HLT ;TYPE IT OUT
1296 004244 000400 BR 1$ ;GET OUT
1297 004246 005724 1$: TST (R4)+ ;GET NEW NUMBER
1298 004250 000760 BR 3$ ;CONTINUE
1299 004252 000240 2$: NOP
  
```

```

1300 ;*****
1301 ;TEST 31 TEST ODD BYTE INSTRUCTIONS ON CS1, CS2, WC AND BA
1302 ;*****
1303 004254 104400 TST31: SCOPE
1304
1305 004256 104414 BITST: CLRDK ;CLEAR ALL RS REG
1306 004260 012777 003566 174616 MOV #3566,@RSCS1 ;LOAD CS1
1307 004266 112777 000005 174644 MOVB #5,@RSCS1B ;LOAD BIT
1308 004274 022777 006766 174602 CMP #6766,@RSCS1 ;DID IT LOAD?
1309 004302 001401 BEQ +4 ;YES
1310 004304 104001 HLT !CS1
1311 004306 112777 000032 174570 MOVB #32,@RSCS1
1312 004314 022777 006632 174562 CMP #6632,@RSCS1
1313 004322 001401 BEQ +4
1314 004324 104001 HLT !CS1 ;CS1 SHOULD = 6632
1315
1316 004326 104400 TST32: SCOPE
1317
1318 004330 013777 001152 174550 BITCS2: MOV UNNUM,@RSCS2 ;LOAD UNIT NUMBER
1319 004336 052777 177400 174542 BIS #177400,@RSCS2 ;LOAD ALL BITS
1320 004344 105077 174572 CLRB @RSCS2B ;CLR UPPER BYTE
1321 004350 013701 001152 MOV UNNUM,GOOD ;GET UNIT NO.
1322 004354 052701 000100 BIS #100,GOOD ;SET OR BIT
1323 004360 017700 174522 MOV @RSCS2,BAD ;GET CS2
1324 004364 020001 CMP BAD,GOOD ;IS CS2 CORRECT?
1325 004366 001401 BEQ +4 ;YES
1326 004370 104000 HLT ;LOAD BYTE DID NOT WORK
1327
1328 004372 104400 TST33: SCOPE
1329
1330 004374 012777 025252 174506 BITWC: MOV #25252,@RSWC ;LOAD WC
1331 004402 112777 000377 174534 MOVB #377,@RSWCB ;LOAD BIT
1332 004410 022777 177652 174472 CMP #177652,@RSWC ;DID IT LOAD?
1333 004416 001401 BEQ +4 ;YES
1334 004420 104010 HLT !WC ;NO WC SHOULD =177652
1335 004422 112777 000123 174460 MOVB #123,@RSWC
1336 004430 022777 177523 174452 CMP #177523,@RSWC
1337 004436 001401 BEQ +4
1338 004440 104010 HLT !WC ;WC SHOULD = 177523
1339
1340 004442 104400 TST34: SCOPE
1341
1342 004444 012777 025252 174440 BITBA: MOV #25252,@RSBA ;LOAD DA
1343 004452 112777 000377 174466 MOVB #377,@RSBAB ;LOAD BIT
1344 004460 022777 177652 174424 CMP #177652,@RSBA ;DID IT LOAD?
1345 004466 001401 BEQ +4 ;YES
1346 004470 104020 HLT !BA ;DA SHOULD =177652
1347 004472 112777 000125 174412 MOVB #125,@RSBA
1348 004500 022777 177524 174404 CMP #177524,@RSBA
1349 004506 001401 BEQ +4
1350 004510 104020 HLT !BA ;BA SHOULD = 177525
1351 004512 104414 CLRDK ;CLEAR ALL RS REG
    
```

```

1352 :*****
1353 :TEST 35 TEST DATA LATE IN CS2
1354 :*****
1355 004514 104400 TST35: SCOPE
1356
1357 ;DO A READ FROM SILO: SHOULD GET DLT + TRE ERROR BECAUSE SILO IS EMPTY
1358
1359 004516 104414 SILOB: CLRDK ;CLEAR ALL RS REG
1360 004520 017700 174402 MOV @RSDB,BAD ;READ FROM EMPTY SILO
1361 004524 017700 174356 MOV @RSCS2,BAD ;GET CS2
1362 004530 012701 100100 MOV #100100,GOOD ;GET CORRECT ANS
1363 004534 053701 001152 BIS UNNUM,GOOD ;FOR CS2
1364 004540 020001 CMP BAD,GOOD ;IS CS2 CORRECT?
1365 004542 001401 BEQ .+4 ;YES
1366 004544 104200 HLT !CS2 ;SHOULD HAVE DLT ERROR
1367 004546 022777 144200 174330 CMP #144200,@RSCS1 ;DID SC AND TRE SET?
1368 004554 001401 BEQ .+4 ;YES
1369 004556 104001 HLT !CS1 ;SC AND TRE SHOULD BE SET
1370 004560 012777 040000 174316 MOV #TRE,@RSCS1 ;CLEARS ERROR BIT
1371 004566 032777 140000 174310 BIT #140000,@RSCS1 ;DID SC + TRE CLEAR
1372 004574 001401 BEQ .+4 ;YES
1373 004576 104001 HLT !CS1 ;TRE AND SC SHOULD BE 0
1374 004600 017700 174302 MOV @RSCS2,BAD ;GET CS2
1375 004604 042701 100000 BIC #BIT15,GOOD ;GET CORRECT ANS
1376 004610 020100 CMP GOOD,BAD ;IS CS2 CORRECT?
1377 004612 001401 BEQ .+4 ;YES
1378 004614 104200 HLT !CS2 ;DLT SHOULD BE 0
1379 :*****
1380 :TEST 36 LOAD RSDB WITH ALL ONES AND ALL ZEROS
1381 :*****
1382 004616 104400 TST36: SCOPE
1383
1384 004620 104414 ZERONE: CLRDK ;CLEAR ALL RS REG
1385 004622 005077 174300 CLR @RSDB ;LOAD DB WITH ALL 0
1386 004626 012777 177777 174272 MOV #177777,@RSDB ;LOAD DB WITH ALL ONES
1387 004634 012737 002000 001174 MOV #2000,WORK ;TIME OUT ROUTINE
1388 004642 012701 000300 MOV #300,GOOD ;GET CORRECT FOR CS2
1389 004646 053701 001152 BIS UNNUM,GOOD
1390 004652 017700 174230 2$: MOV @RSCS2,BAD ;GET CS2
1391 004656 020100 CMP GOOD,BAD ;IS IT CORRECT?
1392 004660 001404 BEQ 3$ ;YES
1393 004662 005337 001174 DEC WORK ;TO WAIT FOR OR
1394 004666 001371 BNE 2$ ;TO SET
1395 004670 104200 HLT !CS2 ;OR SHOULD BE SET
1396 004672 005001 3$: CLR GOOD ;LOAD BAD WITH DB
1397 004674 017700 174226 MOV @RSDB,BAD ;IS BAD CORRECT
1398 004700 020100 CMP GOOD,BAD ;YES
1399 004702 001401 BEQ .+4 ;COULD NOT FLOAT 0 THROUGH DB
1400 004704 104000 HLT ;LOAD GOOD WITH ANS
1401 004706 012701 177777 MOV #-1,GOOD ;GET DATA FROM DB
1402 004712 017700 174210 MOV @RSDB,BAD ;IS DB CORRECT
1403 004716 020100 CMP GOOD,BAD ;YES
1404 004720 001401 BEQ .+4 ;BAD SHOULD = 177777
1405 004722 104000 HLT
    
```

```

1406 004724 104400      TST37: SCOPE
1407                   ;TEST FOR 66 LOCATIONS IN SILO PUT COUNT IN EVERY LOCATION
1408
1409 004726 104414      SILO:  CLRDK                   ;CLEAR ALL RS REG
1410 004730 005001      CLR      R1                   ;CLEAR COUNTER
1411 004732 005201      1$:  INC      R1                   ;INCREMENT COUNTER
1412 004734 010177 174166  MOV     R1,@RSDB              ;LOAD SILO
1413 004740 022701 000102  CMP     #66.,R1              ;LAST LOC. YET?
1414 004744 001372      BNE     1$                   ;NO LOOP AGAIN
1415 004746 012701 000200  MOV     #200,GOOD            ;GET CORRECT ANS FOR CS2
1416 004752 053701 001152  BIS     UNNUM,GOOD
1417 004756 017700 174124  MOV     @RSCS2,BAD           ;GET CS2
1418 004762 020100      CMP     GOOD,BAD            ;IS CS2 CORRECT?
1419 004764 001401      BEQ     .+4                  ;YES
1420 004766 104200      HLT     !CS2                ;OR SHOULD BE 1
1421 004770 005001      CLR     GOOD                ;CLEAR LOCATION COUNTER
1422 004772 005201      2$:  INC     GOOD              ;ADD 1 TO IT
1423 004774 022701 000103  CMP     #67.,GOOD           ;LAST LOC YET?
1424 005000 001405      BEQ     3$                   ;YES
1425 005002 017700 174120  MOV     @RSDB,BAD           ;GET LOC FROM DB
1426 005006 020100      CMP     GOOD,BAD            ;DO LOCATIONS MATCH?
1427 005010 001770      BEQ     2$                   ;YES
1428 005012 104000      HLT     ;CAN NOT MATCH 66 LOCATIONS
1429 005014 032777 000200 174064 3$:  BIT     #OR,@RSCS2          ;IS OR 0
1430 005022 001401      BEQ     .+4                  ;YES
1431 005024 104200      HLT     !CS2                ;OR SHOULD BE 0
1432
1433                   ;NOW PUT 67 WORDS INTO SILO AND CHECK FOR DLT ERROR
1434
1435 005026 005001      4$:  CLR     R1                   ;CLEAR COUNTER
1436 005030 005201      INC     R1                   ;ADD 1 TO COUNT
1437 005032 010177 174070  MOV     R1,@RSDB              ;PUT INTO COUNTER
1438 005036 022701 000103  CMP     #67.,R1              ;DONE YET?
1439 005042 001401      BEQ     .+4                  ;YES
1440 005044 000771      BR      4$                   ;NO DO AGAIN
1441 005046 032777 100000 174032  BIT     #DLT,@RSCS2          ;DID DATA LATE SET?
1442 005054 001001      BNE     .+4                  ;YES
1443 005056 104200      HLT     !CS2                ;DLT DID NOT SET
1444
1445                   ;DOES SILO CHANGE WITH 67TH WORD: IT SHOULD NOT
1446
1447 005060 017700 174042      MOV     @RSDB,BAD           ;GET 1ST WD FORM SILO
1448 005064 012701 000001      MOV     #1,GOOD             ;CORRECT ANS OF SILO
1449 005070 020100      CMP     GOOD,BAD            ;IS SILO GOOD
1450 005072 001401      BEQ     .+4                  ;YES
1451 005074 104000      HLT     ;SILO SHOULD NOT HAVE MOVED
1452 005076 104400      TST40: SCOPE
    
```

```

1453 ;FLOAT A 1 AND A 0 THROUGH THE SILO
1454
1455 005100 104414 SILOFL: CLRDK ;CLEAR ALL RS REG
1456 005102 000241 CLC ;CLEAR CARRY TO FLOAT A 0
1457 005104 012701 000001 MOV #1,GOOD ;GET UP DATA FOR INPUT TO SILO
1458 005110 010177 174012 1$: MOV GOOD,@RSDB ;LOAD DB
1459 005114 006101 ROL GOOD ;SHIFT BIT
1460 005116 103401 BCS .+4 ;DONE YET SHIFTING?
1461 005120 000773 BR 1$ ;NO
1462 005122 012701 177776 MOV #-2,GOOD ;SET ALL ONES
1463 005126 000261 SEC ;SET CARRY TO ROL
1464 005130 010177 173772 3$: MOV GOOD,@RSDB ;LOAD SILO
1465 005134 006101 ROL GOOD ;SHIFT 0
1466 005136 103774 BCS 3$ ;LOOP TILL DONE
1467
1468 ;NOW TEST OUTPUT
1469
1470 005140 000241 CLC ;CLEAR CARRY
1471 005142 012701 000001 MOV #1,GOOD ;CORRECT ANS
1472 005146 017700 173754 2$: MOV @RSDB,BAD ;GET DATA FROM DB
1473 005152 020100 CMP GOOD,BAD ;IS DB DATA GOOD?
1474 005154 001401 BEQ .+4 ;YES
1475 005156 104000 HLT ;DB COULD NOT BUBBLE CORRECTLY
1476 005160 006101 ROL GOOD ;SETUP FOR NEXT ANS
1477 005162 103401 BCS .+4 ;DONE YET?
1478 005164 000770 BR 2$ ;NO
1479 005166 012701 177776 MOV #-2,GOOD ;SETUP FOR ANS
1480 005172 017700 173730 4$: MOV @RSDB,BAD ;GET DATA FROM DB
1481 005176 020100 CMP GOOD,BAD ;IS IT CORRECT?
1482 005200 001401 BEQ .+4 ;YES
1483 005202 104000 HLT ;DB WRONG
1484 005204 000261 SEC ;SET CARRY TO ROL
1485 005206 006101 ROL GOOD ;SETUP FOR NEXT ANS
1486 005210 103770 BCS 4$ ;LOOP TILL DONE
  
```

```

1487
1488
1489
1490 005212 104400
1491
1492 005214 104414
1493 005216 012737 177777 027454
1494 005224 013777 001102 173660
1495 005232 012777 177777 173650
1496 005240 012777 000001 173636
1497 005246 032777 000001 173630
1498 005254 001401
1499 005256 104001
1500 005260 005777 173634
1501 005264 001401
1502 005266 104002
1503 005270 022777 177777 173612
1504 005276 001401
1505 005300 104010
1506 005302 005777 173606
1507 005306 001401
1508 005310 104004
1509 005312 023777 001102 173572
1510 005320 001401
1511 005322 104020
1512 005324 033777 001156 173570
1513 005332 001401
1514 005334 104100

:*****
:TEST 41 TEST NO-OP FUNCTION
:*****
TST41: SCOPE

NOOP: CLRDK ;CLEAR ALL RS REG
MOV #177777,OUTBUF ;DATA TO BE XFERED
MOV @#OBUFSV,@RSBA ;SET UP CURRENT ADDRESS
MOV #-1,@RSWC ;LOAD WC WITH -1
MOV #1,@RSCS1 ;DO NO-OP FUNCTION
BIT #1,@RSCS1 ;DID GO BIT CLEAR
BEQ .+4 ;YES
HLT !CS1 ;GO BIT SHOULD BE CLEARED
TST @RSER ;DID ANY ERRORS OCCUR?
BEQ .+4 ;NO
HLT !ER ;ALL ERROR BITS SHOULD BE 0
CMP #-1,@RSWC ;DID WC MOVE?
BEQ .+4 ;NO
HLT !WC ;WC SHOULD = 177777
TST @RSDA ;DID DA MOV
BEQ .+4 ;NO
HLT !DA ;DA SHOULD =0
CMP @#OBUFSV,@RSBA ;DID BA MOVE
BEQ .+4 ;NO
HLT !BA ;BA MOVED
BIT UNCMP,@RSAS ;AS SHOULD NOT SET ON
BEQ .+4 ;A NO-OP FUNCTION
HLT !AS ;AS SET WHY?
  
```

```

1515 :*****
1516 :TEST 42 TEST NO-OP FUNCTION WITH ERROR BITS SET
1517 :*****
1518 005336 104400 TST42: SCOPE
1519
1520 005340 104414 NNOOP: CLRDK ;CLEAR ALL RS REG
1521 005342 012777 000007 173550 MOV #7,@RSER ;LOAD ER
1522 005350 033777 001156 173544 BIT UNCMP,@RSAS ;IS ATA BIT SET?
1523 005356 001001 BNE .+4 ;YES
1524 005360 104100 HLT !AS ;AS BIT SHOULD BE SET
1525 005362 012737 177777 027454 MOV #177777,OUTBUF ;DATA TO BE XFERED
1526 005370 013777 001102 173514 MOV @#OBUFSV,@RSBA ;SET UP CURRENT ADDRESS
1527 005376 012777 177777 173504 MOV #-1,@RSWC ;LOAD WC WITH -1
1528 005404 012777 000001 173472 MOV #1,@RSCS1 ;DO NO-OP FUNCTION
1529 005412 032777 000001 173464 BIT #1,@RSCS1 ;DID GO BIT CLEAR
1530 005420 001401 BEQ .+4 ;YES
1531 005422 104001 HLT !CS1 ;GO BIT SHOULD BE CLEARED
1532 005424 022777 150600 173464 CMP #150600,@RSDS ;DID ERR BITS SET?
1533 005432 001401 BEQ .+4 ;NO
1534 005434 104040 HLT !DS ;ERR BIT SHOULD BE 0
1535 005436 022777 177777 173444 CMP #-1,@RSWC ;DID WC MOVE?
1536 005444 001401 BEQ .+4 ;NO
1537 005446 104010 HLT !WC ;WC SHOULD = 1777777
1538 005450 005777 173440 TST @RSDA ;DID DA MOV
1539 005454 001401 BEQ .+4 ;NO
1540 005456 104004 HLT !DA ;DA SHOULD =0
1541 005460 023777 001102 173424 CMP @#OBUFSV,@RSBA ;DID BA MOVE
1542 005466 001401 BEQ .+4 ;NO
1543 005470 104020 HLT !BA ;BA MOVED
1544 005472 033777 001156 173422 BIT UNCMP,@RSAS ;AS SHOULD BE SET
1545 005500 001001 BNE .+4 ;IS IT?
1546 005502 104100 HLT !AS ;NO
1547 005504 022777 000007 173406 CMP #7,@RSER ;DID ER CHANGE?
1548 005512 001401 BEQ .+4 ;NO
1549 005514 104002 HLT !ER ;ER SHOULD NOT CHANGE
  
```



```

1550 :*****
1551 :TEST 43 TEST NO-OP FUNCTION CODE 21
1552 :*****
1553 005516 104400 TST43: SCOPE
1554
1555 005520 104414 NOOP21: CLRDK :CLEAR ALL RS REG
1556 005522 012737 177777 027454 MOV #177777,OUTBUF :DATA TO BE XFERED
1557 005530 013777 001102 173354 MOV @#OBUFSV,@RSBA :SET UP CURRENT ADDRESS
1558 005536 012777 177777 173344 MOV #-1,@RSWC :LOAD WC WITH -1
1559 005544 012777 000021 173332 MOV #21,@RSCS1 :DO NO-OP FUNCTION
1560 005552 032777 000001 173324 BIT #1,@RSCS1 :DID GO BIT CLEAR
1561 005560 001401 BEQ .+4 :YES
1562 005562 104001 HLT !CS1 :GO BIT SHOULD BE CLEARED
1563 005564 005777 173330 TST @RSER :DID ANY ERRORS OCCUR?
1564 005570 001401 BEQ .+4 :NO
1565 005572 104002 HLT !ER :ALL ERROR BITS SHOULD BE 0
1566 005574 022777 177777 ;173306 CMP #-1,@RSWC :DID WC MOVE?
1567 005602 001401 BEQ .+4 :NO
1568 005604 104010 HLT !WC :WC SHOULD = 177777
1569 005606 005777 173302 TST @RSDA :DID DA MOV
1570 005612 001401 BEQ .+4 :NO
1571 005614 104004 HLT !DA :DA SHOULD =0
1572 005616 023777 001102 173266 CMP @#OBUFSV,@RSBA :DID BA MOVE
1573 005624 001401 BEQ .+4 :NO
1574 005626 104020 HLT !BA :BA MOVED
1575 005630 033777 001156 173264 BIT UNCMP,@RSAS :AS SHOULD NOT SET ON
1576 005636 001401 BEQ .+4 :A NO-OP FUNCTION
1577 005640 104100 HLT !AS :AS SET WHY?
1578 005642 022777 004220 173234 CMP #4220,@RSCS1 :IS CS1 CORRECT?
1579 005650 001401 BEQ .+4 :YES
1580 005652 104040 HLT !DS :CS1 SHOULD = 4220
    
```

```

1581 :*****
1582 :TEST 44 TEST NO-OP FUNCTION CODE 21 WITH ERROR BITS SET
1583 :*****
1584 005654 104400 TST44: SCOPE
1585
1586 005656 104414 NNOP21: CLRDK ;CLEAR ALL RS REG
1587 005660 012777 000007 173232 MOV #7,@RSER ;LOAD ER
1588 005666 033777 001156 173226 BIT UNCMP,@RSAS ;IS ATA BIT SET?
1589 005674 001001 BNE .+4 ;YES
1590 005676 104100 HLT !AS ;AS BIT SHOULD BE SET
1591 005700 012737 177777 027454 MOV #177777,OUTBUF ;DATA TO BE XFERED
1592 005706 013777 001102 173176 MOV @#OBUFSV,@RSBA ;SET UP CURRENT ADDRESS
1593 005714 012777 177777 173166 MOV #-1,@RSWC ;LOAD WC WITH -1
1594 005722 012777 000021 173154 MOV #21,@RSCS1 ;DO NO-OP FUNCTION
1595 005730 032777 000001 173146 BIT #1,@RSCS1 ;DID GO BIT CLEAR
1596 005736 001401 BEQ .+4 ;YES
1597 005740 104001 HLT !CS1 ;GO BIT SHOULD BE CLEARED
1598 005742 022777 150600 173146 CMP #150600,@RSDS ;DID ERR BITS SET?
1599 005750 001401 BEQ .+4 ;NO
1600 005752 104040 HLT !DS ;ERR BIT SHOULD BE 0
1601 005754 022777 177777 173126 CMP #-1,@RSWC ;DID WC MOVE?
1602 005762 001401 BEQ .+4 ;NO
1603 005764 104010 HLT !WC ;WC SHOULD = 1777777
1604 005766 005777 173122 TST @RSDA ;DID DA MOV
1605 005772 001401 BEQ .+4 ;NO
1606 005774 104004 HLT !DA ;DA SHOULD =0
1607 005776 023777 001102 173106 CMP @#OBUFSV,@RSBA ;DID BA MOVE
1608 006004 001401 BEQ .+4 ;NO
1609 006006 104020 HLT !BA ;BA MOVED
1610 006010 033777 001156 173104 BIT UNCMP,@RSAS ;AS SHOULD BE SET
1611 006016 001001 BNE .+4 ;IS IT?
1612 006020 104100 HLT !AS ;NO
1613 006022 022777 000007 173070 CMP #7,@RSER ;DID ER CHANGE?
1614 006030 001401 BEQ .+4 ;NO
1615 006032 104002 HLT !ER ;ER SHOULD NOT CHANGE
1616 006034 022777 104220 173042 CMP #104220,@RSCS1 ;IS CS1 CORRECT?
1617 006042 001401 BEQ .+4 ;YES
1618 006044 104040 HLT !DS ;CS1 SHOULD = 104220
    
```

```

1619 :*****
1620 :TEST 45 TEST DRIVE CLEAR FUNCTION WITH ERRORS SET
1621 :*****
1622 006046 104400 TST45: SCOPE
1623 :FIRST SET ALL R/W BITS IN DISK REG
1624 :DO DRIVE CLEAR-ALL R/W BITS SHOULD BE CLEARED
1625
1626 006050 104414 DRCLR: CLRDK ;CLEAR ALL RS REG
1627 006052 012777 177777 173030 MOV #177777,@RSWC ;LOAD RSWC
1628 006060 012777 177777 173026 MOV #177777,@RSDA ;SET ALL POSSIBLE
1629 006066 012777 177017 173024 MOV #177017,@RSER ;BITS IN DISK REG
1630 006074 012777 000070 173026 MOV #70,@RSMR ;SET THESE BITS
1631 006102 012777 000011 172774 MOV #11,@RSCS1 ;SET DRIVE CLEAR
1632 006110 017700 172772 MOV @RSCS2,BAD ;GET CS2 DATA
1633 006114 042700 177640 BIC #177640,BAD ;CLEAR JUNK
1634 006120 013701 001152 MOV UNNUM,GOOD ;GET DRIVE UNIT
1635 006124 052701 000100 BIS #100,GOOD ;SET IR BIT
1636 006130 020100 CMP GOOD,BAD ;IS UNIT # THE SAME
1637 006132 001401 BEQ .+4 ;YES
1638 006134 104000 HLT ;UNIT # IN CS2 GOT MODIFIED
1639 006136 005777 172752 TST @RSDA ;DID DA CLEAR
1640 006142 001401 BEQ .+4 ;YES
1641 006144 104004 HLT !DA ;DA SHOULD BE 0
1642 006146 005777 172746 TST @RSER ;DID ER CLEAR
1643 006152 001401 BEQ .+4 ;YES
1644 006154 104002 HLT !ER ;ER SHOULD BE CLEARED
1645 006156 017737 172746 001174 MOV @RSMR,WORK ;GET MR REG
1646 006164 042737 177707 001174 BIC #177707,WORK ;CLEAR JUNK
1647 006172 022737 000070 001174 CMP #70,WORK ;IS 70 STILL SET IN MR?
1648 006200 001401 BEQ .+4 ;YES
1649 006202 104220 HLT !MR ;BITS 70 SHOULD NOT CLEAR
1650 006204 022777 004210 172672 CMP #4210,@RSCS1 ;DID THESE BITS CLEAR?
1651 006212 001401 BEQ .+4 ;YES
1652 006214 104001 HLT !CS1 ;CS1 SHOULD =4210
1653 006216 033777 001156 172676 BIT UNCMP,@RSAS ;AS SHOULD NOT SET
1654 006224 001401 BEQ .+4 ;ON A DRIVE CLEAR FUN
1655 006226 104100 HLT !AS ;WHY DID AS SET?
1656 006230 022777 177777 172652 CMP #177777,@RSWC ;DID RSWC CHANGE?
1657 006236 001401 BEQ .+4 ;NO
1658 006240 104010 HLT !WC
  
```

```

1659 ;DO ONE WORD WRITE
1660 ;*****
1661 ;TEST 46 EXECUTE THE ONE WORD WRITE
1662 ;*****
1663 006242 104400 TST46: SCOPE
1664
1665 006244 104414 WRTST: CLRDK ;CLEAR ALL RS REG
1666 006246 012737 177777 027454 MOV #177777,OUTBUF ;DATA TO BE X-FERED
1667 006254 013777 001102 172630 MOV @#OBUFSV,@RSBA ;SET UP CURRENT ADDRESS
1668 006262 012777 177777 172620 MOV #-1,@RSWC ;SET WORD COUNT TO -1
1669 006270 012777 000060 172606 1$: MOV #60,@RSCS1 ;SET FUNCTION WITH NO GO BIT
1670 006276 022777 177777 172604 CMP #-1,@RSWC ;DID WC MOVE?
1671 006304 001401 BEQ .+4 ;NO
1672 006306 104010 HLT !WC ;WC MOVED
1673 006310 023777 001102 172574 CMP @#OBUFSV,@RSBA ;DID RSBA MOVE?
1674 006316 001401 BEQ .+4 ;NO
1675 006320 104020 HLT !BA ;BA MOVED
1676 006322 052777 000001 172554 BIS #BIT0,@RSCS1 ;SET GO BIT
1677 006330 105777 172550 2$: TSTB @RSCS1 ;TEST FOR RDY=0
1678 006334 100001 BPL .+4 ;RDY=0
1679 006336 104001 HLT !CS1 ;RDY SHOULD = 0
1680 006340 004737 026670 JSR PC,WAITRY ;WAIT FOR READY
1681 006344 104001 HLT !CS1 ;SHOULD = 260 RDY NEVER CAME UP
1682 006346 022777 000001 172540 CMP #1,@RSDA ;IS RSDA CORRECT
1683 006354 001401 BEQ .+4 ;RSDA OK
1684 006356 104004 HLT !DA ;SHOULD = 1 SHOULD INCREMENT
1685 006360 022777 004260 172516 3$: CMP #4260,@RSCS1 ;IS ERROR FLAG SET?
1686 006366 001401 BEQ .+4 ;NO! X-FER OK
1687 006370 104047 HLT !CS1!ER!DS!DA ;ERROR DURING X-FER
1688 006372 005777 172512 4$: TST @RSWC ;FETCH WORD COUNT
1689 006376 001401 BEQ .+4 ;WORD COUNT DID OVERFLOW
1690 006400 104010 HLT !WC ;SHOULD = 0 FAILED TO INCREMENT
1691 006402 022777 010600 172506 CMP #10600,@RSDS ;IS RSDS OK?
1692 006410 001401 BEQ .+4 ;YES
1693 006412 104044 HLT !DS!DA ;NO
1694 006414 013701 001152 MOV UNNUM,GOOD ;GET UNIT #
1695 006420 052701 000100 BIS #100,GOOD ;SET IR BIT
1696 006424 017700 172450 MOV @RSCS2,BAD ;GET CS2
1697 006430 020100 CMP GOOD,BAD ;IS CS2 CORRECT?
1698 006432 001401 BEQ .+4 ;YES
1699 006434 104000 HLT ;BAD = CS2 GOOD IS CORRECT ANS
1700 006436 017700 172450 MOV @RSBA,BAD ;GET BA DATA
1701 006442 013701 001102 MOV @#OBUFSV,GOOD ;WHAT RSBA SHOULD EQUAL
1702 006446 062701 000002 ADD #2,GOOD ;UPDATE OUTBUFFER
1703 006452 020100 CMP GOOD,BAD ;IS RSBA CORRECT
1704 006454 001401 BEQ .+4 ;YES
1705 006456 104000 HLT ;BA FAILED TO INCREMENT
1706 006460 005777 172434 TST @RSER ;DID ANY ERRORS SET?
1707 006464 001401 BEQ .+4 ;NO
1708 006466 104040 HLT !DS
    
```

```

1709 ;TEST READ FUNCTION
1710
1711 :*****
1712 :TEST 47 EXECUTE THE ONE WORD READ
1713 :*****
1714 006470 104400 TST47: SCOPE
1715
1716 006472 104414 RDTST: CLRDK ;CLEAR ALL RS REG
1717 006474 005037 027454 CLR ;CLR TO READ INTO
1718 006500 013777 001102 172404 MOV @#OBUFSV,@RSBA ;SET UP CURRENT ADDRESS
1719 006506 012777 177777 172374 MOV #-1,@RSCW ;SET WORD COUNT TO -1
1720 006514 012777 000071 172362 1$: MOV #71,@RSCS1 ;GO READ
1721 006522 105777 172356 2$: TSTB @RSCS1 ;TEST FOR BUSY=1
1722 006526 100001 BPL .+4 ;BUSY SET
1723 006530 104001 HLT !CS1 ;BUSY NOT SET
1724 006532 004737 026670 JSR PC,WAIRY ;WAIT FOR READY
1725 006536 104001 HLT !CS1 ;TIMEOUT RDY DID NOT SET
1726 006540 022777 000001 172346 CMP #BIT0,@RSDA ;WAS RSDA INCREMENTED BY 1
1727 006546 001401 BEQ .+4 ;RSDA OK
1728 006550 104046 HLT !DA!ER!DS ;RSDA SHOULD CONTAIN A 1
1729 006552 022777 004270 172324 3$: CMP #4270,@RSCS1 ;IS ERROR FLAG SET?
1730 006560 001401 BEQ .+4 ;NO! X-FER OK
1731 006562 104043 HLT !CS1!ER!DS ;RSCS1 SHOULD = 270
1732 006564 005777 172320 4$: TST @RSCW ;TEST WC
1733 006570 001401 BEQ .+4 ;WORD COUNT DID OVERFLOW
1734 006572 104010 HLT !WC ;SHOULD = 0
1735 006574 013701 001152 MOV UNNUM,GOOD ;GET CORRECT
1736 006600 052701 000100 BIS #100,GOOD ;ANS OF CS2
1737 006604 017700 172276 MOV @RSCS2,BAD ;GET CS2
1738 006610 020100 CMP GOOD,BAD ;IS CS2 CORRECT?
1739 006612 001401 BEQ .+4 ;YES
1740 006614 104000 HLT ;GOOD = CORRECT ANS FOR CS2
1741 006616 017700 172270 MOV @RSBA,BAD ;FETCH CURRENT ADDRESS
1742 006622 013701 001102 MOV @#OBUFSV,GOOD ;WHAT RSBA SHOULD EQUAL
1743 006626 062701 000002 ADD #2,GOOD ;UPDATE IT
1744 006632 020001 CMP BAD,GOOD ;IS RSBA CORRECT
1745 006634 001401 BEQ .+4 ;YES EXECUTE CONTINUE
1746 006636 104000 HLT ;RSBA FAILED TO INCREMENT
1747 006640 013700 027454 MOV OUTBUF,BAD ;GET DATA READ FROM DISK
1748 006644 012701 177777 MOV #-1,GOOD ;GET CORRECT ANS
1749 006650 020100 CMP GOOD,BAD ;IS OUTBUF CORRECT
1750 006652 001401 BEQ .+4 ;YES
1751 006654 104000 HLT ;GOOD=CORRECT ANS BAD=DATA READ FROM DISK
    
```

```

1752 :*****
1753 :TEST 50 TEST WRITE CHECK
1754 :*****
1755 006656 104400 TST50: SCOPE
1756 :DO A ONE WORD WRITE CHECK
1757
1758 :* * *EXECUTE THE ONE WORD WRITE CHECK* * *
1759
1760 006660 104414 WRCKT: CLRDK :CLEAR ALL RS REG
1761 006662 012737 177777 027454 MOV #177777,OUTBUF :DATA TO BE X-FERED
1762 006670 013777 001102 172214 MOV @#OBUFSV,@RSBA :SET UP CURRENT ADDRESS
1763 006676 012777 177777 172204 MOV #-1,@RSWC :SET WORD COUNT TO -1
1764 006704 012777 000051 172172 1$: MOV #51,@RSCS1 :GO WRITE CHECK
1765 006712 105777 172166 2$: TSTB @RSCS1 :TEST FOR READY
1766 006716 100001 BPL .+4 :NOT READY
1767 006720 104001 HLT !CS1 :BUSY FAILED TO SET
1768 006722 004737 026670 RSWCWT: JSR PC,WAITRY :WAIT FOR READY
1769 006726 104001 HLT !CS1 :BUSY FAILED TO CLEAR
1770 006730 013701 001152 MOV UNNUM,GOOD :GET UNIT #
1771 006734 052701 000100 BIS #100,GOOD :SET BIT IR
1772 006740 017700 172142 MOV @RSCS2,BAD :GET CS2
1773 006744 020100 CMP GOOD,BAD :IS CS2 CORRECT?
1774 006746 001401 BEQ .+4 :YES
1775 006750 104000 HLT :GOOD = CORRECT ANS FOR CS2
1776 006752 022777 004250 172124 2$: CMP #4250,@RSCS1 :ANY ERRORS?
1777 006760 001401 BEQ .+4 :X-FER OK
1778 006762 104046 HLT !DA!ER!DS :ERROR DUR X-FER
1779 006764 022777 000001 172122 3$: CMP #BITO,@RSDA :WAS DAR INCREMENTED BY 1
1780 006772 001401 BEQ .+4 :RSDA OK
1781 006774 104004 HLT !DA :DAR SHOULD = 1
1782 006776 005777 172106 TST @RSWC :TEST FOR OVERFLOW
1783 007002 001401 BEQ .+4 :WORD COUNT DID OVERFLOW
1784 007004 104010 HLT !WC :SHOULD = 0
1785 007006 017700 172100 MOV @RSBA,BAD :FETCH CURRENT ADDRESS
1786 007012 013701 001102 MOV @#OBUFSV,GOOD :WHAT RSBA SHOULD EQUAL
1787 007016 062701 000002 ADD #2,GOOD :UPDATE IT
1788 007022 020001 CMP BAD,GOOD :IS RSBA CORRECT
1789 007024 001401 BEQ .+4 :YES EXECUTE CONTINUE
1790 007026 104000 HLT :RSBA FAILED TO INCREMENT
  
```

```

1791 :DO ONE WORD WRITE ON -B- PORT
1792 :IF A 1 WD TRANSFER KEEPS SETTING NEM PROGAM WILL GO AND UPDATE
1793 :ADDRESS (OBUFSV) ON -B- PORT BY 4K AND TRY TRANSFER AGAIN UNTILL IT
1794 :REACHES 28K. IF NO TRANSFER IT THEN SKIPS WRITE,
1795 :READ AND WRITE CHECK ON -B- PORT
1796 :TO INHIBIT OBUFSV FROM CHANGING SET BIT 12
1797
1798 :*****
1799 :TEST 51 EXECUTE THE ONE WORD WRITE ON -B- PORT
1800 :*****
1801 007030 104400 TST51: SCOPE
1802
1803 007032 104414 WRTSTB: CLRDK ;CLEAR ALL RS REG
1804 007034 013737 001102 001174 MOV @#OBUFSV,WORK ;GET LOC OF RSBA TO LOAD
1805 007042 012777 177777 172124 MOV #177777,@WORK ;DATA TO BE X-FERED
1806 007050 013777 001102 172034 MOV @#OBUFSV,@RSBA ;SET UP CURRENT ADDRESS
1807 007056 012777 177777 172024 MOV #-1,@RSWC ;SET WORD COUNT TO -1
1808 007064 012777 002061 172012 MOV #2061,@RSCS1 ;TEST B PORT
1809 007072 105777 172006 2$: TSTB @RSCS1 ;TEST FOR RDY=0
1810 007076 100001 BPL .+4 ;RDY=0
1811 007100 104001 HLT !CS1 ;RDY SHOULD = 0
1812 007102 004737 026670 JSR PC,WAIRY ;WAIT FOR READY
1813 007106 104001 HLT !CS1 ;SHOULD = 260 RDY NEVER CAME UP
1814 007110 032777 010000 171710 BIT #BIT12,@SWR ;INHIBIT ADDRESS?
1815 007116 001006 BNE 3$ ;YES
1816 007120 032777 004000 171760 BIT #BIT11,@RSCS2 ;DID NEM SET?
1817 007126 001402 BEQ 3$ ;NO
1818 007130 000137 025722 JMP FINDM ;GO FIND MEMORY ON PORT B
1819 007134 022777 000001 171752 3$: CMP #1,@RSDA ;IS RSDA CORRECT
1820 007142 001401 BEQ .+4 ;RSDA OK
1821 007144 104004 HLT !DA ;SHOULD = 1 SHOULD INCREMENT
1822 007146 022777 006260 171730 CMP #6260,@RSCS1 ;IS CS1 CORRECT?
1823 007154 001401 BEQ 4$ ;YES
1824 007156 104001 HLT !CS1
1825 007160 005777 171724 4$: TST @RSWC ;FETCH WORD COUNT
1826 007164 001401 BEQ .+4 ;WORD COUNT DID OVERFLOW
1827 007166 104010 HLT !WC ;SHOULD = 0 FAILED TO INCREMENT
1828 007170 022777 010600 171720 CMP #10600,@RSDS ;IS RSDS OK?
1829 007176 001401 BEQ .+4 ;YES
1830 007200 104044 HLT !DS!DA ;NO
1831 007202 013701 001152 MOV UNNUM,GOOD ;GET UNIT #
1832 007206 052701 000100 BIS #100,GOOD ;SET IR BIT
1833 007212 017700 171670 MOV @RSCS2,BAD ;GET CS2
1834 007216 020100 CMP GOOD,BAD ;IS CS2 CORRECT?
1835 007220 001401 BEQ .+4 ;YES
1836 007222 104000 HLT ;BAD = CS2 GOOD IS CORRECT ANS
1837 007224 017700 171662 MOV @RSBA,BAD ;GET BA DATA
1838 007230 013701 001102 MOV @#OBUFSV,GOOD ;WHAT RSBA SHOULD EQUAL
1839 007234 062701 000002 ADD #2,GOOD ;UPDATE OUTBUFFER
1840 007240 020100 CMP GOOD,BAD ;IS RSBA CORRECT
1841 007242 001401 BEQ .+4 ;YES
1842 007244 104000 HLT ;BA FAILED TO INCREMENT
    
```

```

1843 :*****
1844 :TEST 52 EXECUTE THE ONE WORD READ ON -B- PORT
1845 :*****
1846 007246 104400 TST52: SCOPE
1847
1848 007250 104414 RDTSTB: CLRDK ;CLEAR ALL RS REG
1849 007252 005037 001174 CLR WORK ;CLR TO READ INTO
1850 007256 013777 001102 171626 MOV @#OBUFSV,@RSBA ;SET UP CURRENT ADDRESS
1851 007264 012777 177777 171616 MOV #-1,@RSWC ;SET WORD COUNT TO -1
1852 007272 012777 002071 171604 MOV #2071,@RSCS1 ;B PORT
1853 007300 105777 171600 2$: TSTB @RSCS1 ;TEST FOR BUSY=1
1854 007304 100001 BPL .+4 ;BUSY SET
1855 007306 104001 HLT !CS1 ;BUSY NOT SET
1856 007310 004737 026670 JSR PC,WAITRY ;WAIT FOR READY
1857 007314 104001 HLT !CS1 ;TIMEOUT RDY DID NOT SET
1858 007316 022777 000001 171570 CMP #BIT0,@RSDA ;WAS RSDA INCREMENTED BY 1
1859 007324 001401 BEQ .+4 ;RSDA OK
1860 007326 104046 HLT !DA!ER!DS ;RSDA SHOULD CONTAIN A 1
1861 007330 022777 006270 171546 CMP #6270,@RSCS1 ;TST B PORT
1862 007336 001401 BEQ 4$ ;OK
1863 007340 104001 HLT !CS1 ;CS1 SHOULD = 6270
1864 007342 005777 171542 4$: TST @RSWC ;TEST WC
1865 007346 001401 BEQ .+4 ;WORD COUNT DID OVERFLOW
1866 007350 104010 HLT !WC ;SHOULD = 0
1867 007352 013701 001152 MOV UNNUM,GOOD ;GET CORRECT
1868 007356 052701 000100 BIS #100,GOOD ;ANS OF CS2
1869 007362 017700 171520 MOV @RSCS2,BAD ;GET CS2
1870 007366 020100 CMP GOOD,BAD ;IS CS2 CORRECT?
1871 007370 001401 BEQ .+4 ;YES
1872 007372 104000 HLT ;GOOD = CORRECT ANS FOR CS2
1873 007374 017700 171512 MOV @RSBA,BAD ;FETCH CURRENT ADDRESS
1874 007400 013701 001102 MOV @#OBUFSV,GOOD ;WHAT RSBA SHOULD EQUAL
1875 007404 062701 000002 ADD #2,GOOD ;UPDATE IT
1876 007410 020001 CMP BAD,GOOD ;IS RSBA CORRECT
1877 007412 001401 BEQ .+4 ;YES EXECUTE CONTINUE
1878 007414 104000 HLT ;RSBA FAILED TO INCREMENT
1879 007416 013737 001102 001174 MOV @#OBUFSV,WORK ;GET DATA READ FROM DISK
1880 007424 017700 171544 MOV @WORK,BAD
1881 007430 012701 177777 MOV #-1,GOOD ;GET CORRECT ANS
1882 007434 020100 CMP GOOD,BAD ;IS OUTBUF CORRECT
1883 007436 001401 BEQ .+4 ;YES
1884 007440 104000 HLT ;GOOD=CORRECT ANS BAD=DATA READ FROM DISK
  
```



```

1885 :*****
1886 :TEST 53 TEST WRITE CHECK ON -B- PORT
1887 :*****
1888 007442 104400 TST53: SCOPE
1889
1890
1891 007444 104414 WRCKTB: CLRDK ;CLEAR ALL RS REG
1892 007446 013737 001102 001174 MOV @#OBUF SV,WORK ;GET LOC FOR
1893 007454 012777 177777 171512 MOV #177777,@WORK ;DATA TO BE X-FERED
1894 007462 013777 001102 171422 MOV @#OBUF SV,@RSBA ;SET UP CURRENT ADDRESS
1895 007470 012777 177777 171412 MOV #-1,@RSWC ;SET WORD COUNT TO -1
1896 007476 012777 002051 171400 MOV #2051,@RSCS1 ;B PORT
1897 007504 105777 171374 2$: TSTB @RSCS1 ;TEST FOR READY
1898 007510 100001 BPL .+4 ;NOT READY
1899 007512 104001 HLT !CS1 ;BUSY FAILED TO SET
1900 007514 004737 026670 JSR PC,WAIRY ;WAIT FOR READY
1901 007520 104001 HLT !CS1 ;BUSY FAILED TO CLEAR
1902 007522 013701 001152 MOV UNNUM,GOOD ;GET UNIT #
1903 007526 052701 000100 BIS #100,GOOD ;SET BIT IR
1904 007532 017700 171350 MOV @RSCS2,BAD ;GET CS2
1905 007536 020100 CMP GOOD,BAD ;IS CS2 CORRECT?
1906 007540 001401 BEQ .+4 ;YES
1907 007542 104000 HLT ;GOOD = CORRECT ANS FOR CS2
1908 007544 022777 006250 171332 CMP #6250,@RSCS1 ;IS CS1 CORRECT?
1909 007552 001401 BEQ 3$ ;YES
1910 007554 104001 HLT !CS1 ;CS1 SHOULD = 6250
1911 007556 022777 000001 171330 3$: CMP #BIT0,@RSDA ;WAS DAR INCREMENTED BY 1
1912 007564 001401 BEQ .+4 ;RSDA OK
1913 007566 104004 HLT !DA ;DAR SHOULD = 1
1914 007570 005777 171314 TST @RSWC ;TEST FOR OVERFLOW
1915 007574 001401 BEQ .+4 ;WORD COUNT DID OVERFLOW
1916 007576 104010 HLT !WC ;SHOULD = 0
1917 007600 017700 171306 MOV @RSBA,BAD ;FETCH CURRENT ADDRESS
1918 007604 013701 001102 MOV @#OBUF SV,GOOD ;WHAT RSBA SHOULD EQUAL
1919 007610 062701 000002 ADD #2,GOOD ;UPDATE IT
1920 007614 020001 CMP BAD,GOOD ;IS RSBA CORRECT
1921 007616 001401 BEQ .+4 ;YES EXECUTE CONTINUE
1922 007620 104000 HLT ;RSBA FAILED TO INCREMENT
    
```

```

1923 007622 013737 001102 001170 NXM:  MOV  OBUFSV,BPORTT ;SAVE -B- PORT BUFFER
1924 007630 012737 027454 001102      MOV  #OUTBUF ,OBUFSV ;RESTORE OBUFSV
1925
1926 ;DESELECT THEN SELECT UNIT NUMBER IN RSCS2 CHECK TIMING
1927 ;*****
1928 ;TEST 54 DESELECT THEN SELECT UNIT NUMBER TIMING TEST
1929 ;*****
1930 007636 104400 TST54: SCOPE
1931
1932 007640 104414 UNITST: CLRDK ;CLEAR ALL RS REG
1933 007642 005004      CLR  R4 ;CLEAR R4
1934 007644 020437 001152      CMP  R4,UNNUM ;IS THIS CORRECT UNIT #?
1935 007650 001001      BNE  3$ ;NO THEN USE IT
1936 007652 005204      INC  R4 ;GET WRONG DRIVE
1937 007654 012737 177777 027454 3$:  MOV  #177777,OUTBUF ;DATA TO BE X-FERED
1938 007662 013777 001102 171222      MOV  @#OBUFSV,@RSBA ;SET UP CURRENT ADDRESS
1939 007670 012777 177777 171212      MOV  #-1,@RSWC ;SET WORD COUNT TO -1
1940 007676 012703 000061      MOV  #61,R3 ;GET WRITE FUNCTION
1941 007702 013705 001152      MOV  UNNUM,R5 ;GET CORRECT UNIT #
1942 007706 012701 172040      MOV  #172040,R1 ;GET CS1 REG
1943 007712 010461 000010      MOV  R4,10(R1) ;LOAD WRONG UNIT # INTO CS2
1944 007716 000240      NOP ;WAIT FOR DRIVE TO SETTLE
1945 007720 010561 000010      MOV  R5,10(R1) ;LOAD CORRECT UNIT #
1946 007724 010311 1$:  MOV  R3,(R1) ;LOAD FUNCTION IN CS1
1947 007726 004737 026670      JSR  PC,WAITRY ;WAIT FOR READY
1948 007732 104001      HLT  !CS1 ;SHOULD = 260 RDY NEVER CAME UP
1949 007734 022777 004260 171142      CMP  #4260,@RSCS1 ;IS ERROR FLAG SET?
1950 007742 001401      BEQ  .+4 ;NO! X-FER OK
1951 007744 104047      HLT  !CS1!ER!DS!DA ;ERROR DURING X-FER
1952 007746 022777 000001 171140      CMP  #1,@RSDA ;IS RSDA CORRECT
1953 007754 001401      BEQ  .+4 ;RSDA OK
1954 007756 104004      HLT  !DA ;SHOULD = 1 SHOULD INCREMENT
1955 007760 005777 171124      TST  @RSWC ;FETCH WORD COUNT
1956 007764 001401      BEQ  .+4 ;WORD COUNT DID OVERFLOW
1957 007766 104010      HLT  !WC ;SHOULD = 0 FAILED TO INCREMENT
1958 007770 022777 010600 171120      CMP  #10600,@RSDS ;IS RSDS OK?
1959 007776 001401      BEQ  .+4 ;YES
1960 010000 104044      HLT  !DS!DA ;NO
1961 010002 013701 001152      MOV  UNNUM,GOOD ;GET UNIT #
1962 010006 052701 000100      BIS  #100,GOOD ;SET IR BIT
1963 010012 017700 171070      MOV  @RSCS2,BAD ;GET CS2
1964 010016 020100      CMP  GOOD,BAD ;IS CS2 CORRECT?
1965 010020 001401      BEQ  .+4 ;YES
1966 010022 104000      HLT ;BAD = CS2 GOOD IS CORRECT ANS
1967 010024 017700 171062      MOV  @RSBA,BAD ;GET BA DATA
1968 010030 013701 001102      MOV  @#OBUFSV,GOOD ;WHAT RSBA SHOULD EQUAL
1969 010034 062701 000002      ADD  #2,GOOD ;UPDATE OUTBUFFER
1970 010040 020100      CMP  GOOD,BAD ;IS RSBA CORRECT
1971 010042 001401      BEQ  .+4 ;YES
1972 010044 104000      HLT ;BA FAILED TO INCREMENT
    
```

1973  
 1974  
 1975  
 1976  
 1977  
 1978  
 1979  
 1980 010046 104400  
 1981  
 1982 010050 104414  
 1983 010052 013777 001102 171032  
 1984 010060 012777 177777 171022  
 1985 010066 052777 000010 171012  
 1986 010074 012777 000061 171002  
 1987 010102 004737 026670  
 1988 010106 104001  
 1989 010110 013701 001102  
 1990 010114 017700 170772  
 1991 010120 020100  
 1992 010122 001401  
 1993 010124 104000  
 1994 010126 005777 170766  
 1995 010132 001401  
 1996 010134 104040  
 1997 010136 104414  
 1998 010140 032777 000010 170740  
 1999 010146 001401  
 2000 010150 104002

```

:TEST CURRENT ADDRESS INHIBT-BAI IN RSCS2
:DO A ONE WORD WRITE AND SEE
:IF RSBA INCREMENTED AFTER THE X-FER

:*****
:TEST 55 TEST BAI IN RSCS2
:*****
TST55: SCOPE

BAITST: CLRDK ;CLEAR ALL RS REG
MOV @#OBUF SV,@RSBA ;SET UP CURRENT ADDR
MOV #-1,@RSWC ;SET WORD COUNT TO -1
BIS #BAI,@RSCS2 ;SET BAI BIT
MOV #61,@RSCS1 ;WRITE
JSR PC, WAITRY ;WAIT FOR READY
HLT !CS1 ;RDY DID NOT SET
$: MOV @#OBUF SV,GOOD ;WHAT RSBA SHOULD BE
MOV @RSBA,BAD ;WHAT RSBA IS
CMP GOOD,BAD ;COMPARE
BEQ .+4 ;YES
HLT ;BAD=OUTBUF GOOD = CORRECT ANS
TST @RSER ;ANY ERRORS?
BEQ .+4 ;NO
HLT !DS ;YES
CLRDK ;CLEAR ALL RS REG
BIT #BAI,@RSCS2 ;DID BAI CLEAR?
BEQ .+4 ;YES
HLT !ER ;BAI DID NOT SET
    
```

```

2001 :*****
2002 :TEST 56 TEST NON-EXISTENT MEMORY ERROR BIT IN CS2
2003 :*****
2004 010152 104400 TST56: SCOPE
2005
2006 010154 104414 NXMTSM: CLRDK :CLEAR ALL RS REG
2007 010156 052777 000010 170722 BIS #BAI,@RSCS2 :SET BAI BIT
2008 010164 012777 177600 170716 MOV #-200,@RSWC :SET UP WORD COUNT
2009 010172 012777 173000 170712 MOV #173000,@RSBA :SET UP CURRENT ADDRESS
2010 010200 012777 001471 170676 MOV #1471,@RSCS1 :READ AND LOAD A16 +A17 FOR 18 BIT ADDRESS
2011 010206 004737 026670 JSR PC,WAITRY :WAIT FOR READY
2012 010212 104040 HLT !DS :READY NEVER CAME UP
2013 010214 013701 001152 TSTNEM: MOV UNNUM,GOOD :GET UNIT NO.
2014 010220 052701 004310 BIS #4310,GOOD :SET BAI+OR BITS
2015 010224 017700 170656 MOV @RSCS2,BAD :GET CS2
2016 010230 020100 CMP GOOD,BAD :IS CS2 CORRECT?
2017 010232 001401 BEQ .+4 :YES
2018 010234 104000 HLT :BAD=CS2 GOOD=CORRECT ANS FOR CS2
2019 010236 022777 145670 170640 CMP #145670,@RSCS1 :DID TRE SET?
2020 010244 001401 BEQ .+4 :YES
2021 010246 104001 HLT !CS1 :TRE SHOULD SET BECAUSE OF NEM
2022 010250 012777 040000 170626 MOV #TRE,@RSCS1 :CLEAR TRE
2023 010256 017700 170624 MOV @RSCS2,BAD :GET CS2
2024 010262 013701 001152 MOV UNNUM,GOOD :GET DRIVE
2025 010266 052701 000310 BIS #310,GOOD :SET IR
2026 010272 020100 CMP GOOD,BAD :IS CS2 CORRECT?
2027 010274 001401 BEQ .+4 :YES
2028 010276 104200 HLT !CS2 :CS2=BAD GOOD IS CORRECT ANS FOR CS2
2029

```

```

2030 :*****
2031 :TEST 57 TEST BLOCK SEARCH FUNCTION, PIP AND DRY BIT AND ADDR. CONF BIT
2032 :*****
2033 010300 104400 TST57: SCOPE
2034
2035 010302 104414 BLOCK: CLRDK :CLEAR ALL RS REG
2036 010304 012777 000032 170602 MOV #32,@RSDA :DO A SEARCH FOR SECTOR 32
2037 010312 013777 001102 170572 MOV @#0BUFSV,@RSBA :LOAD REGS. TO MAKE
2038 010320 012777 177777 170562 MOV #-1,@RSWC :SURE THEY DO NOT CHANGE
2039 010326 005037 001174 CLR WORK :SETUP FOR TIMEOUT ROUTINE
2040 010332 032777 001000 170570 4$: BIT #1000,@RSMR :WAIT FOR DISK TO
2041 010340 001004 BNE 3$ :REACH SECTOR 32
2042 010342 005337 001174 DEC WORK :TIME OUT
2043 010346 001371 BNE 4$ :ROUTINE
2044 010350 104220 HLT :MR :COULD NOT FIND SECTOR 32

```

```

2045 010352 005077 170536 3$: CLR @RSDA ;NOW SEARCH FOR 0
2046 010356 012777 000031 170520 MOV #31,@RSCS1 ;DO A BLOCK SEARCH FUNCTION
2047 010364 032777 000200 170524 BIT #DRY,@RSDS ;IS DRY CLEARED?
2048 010372 001402 BEQ 1$ ;YES
2049 010374 104040 HLT !DS ;DRY SHOULD BE CLEARED DURING A BLOCK SEARCH
2050 010376 000500 BR OOUT ;GET OUT BECAUSE OF TIMING
2051 010400 032777 020000 170510 1$: BIT #20000,@RSDS ;IS PIP SET?
2052 010406 001001 BNE .+4 ;YES
2053 010410 104040 HLT !DS ;PIP SHOULD BE SET
2054 010412 012701 020000 MOV #20000,GOOD ;SETUP FOR TIMEOUT ROUTINE
2055 010416 005301 2$: DEC GOOD ;DO TIMEOUT
2056 010420 001466 BEQ TTMOU ;TIMED OUT
2057 010422 032777 000200 170466 BIT #DRY,@RSDS ;DID DRY SET?
2058 010430 001772 BEQ 2$ ;NO
2059 010432 022777 110600 170456 CMP #110600,@RSDS ;DID PIP CLEAR?
2060 010440 001401 BEQ .+4 ;YES
2061 010442 104040 HLT !DS ;PIP BIT DID NOT CLEAR
2062 010444 022777 104230 170432 CMP #104230,@RSCS1 ;DID SC SET?
2063 010452 001401 BEQ .+4 ;YES
2064 010454 104041 HLT !CS1!DS ;SC DID NOT SET
2065 010456 013737 001152 001174 MOV UNNUM,WORK ;GET CORRECT AS BIT
2066 010464 005001 CLR GOOD ;IN RSAS REG
2067 010466 000261 SEC ;THAT SHOULD
2068 010470 006101 5$: ROL GOOD ;BE SET
2069 010472 005737 001174 TST WORK
2070 010476 001403 BEQ 6$
2071 010500 005337 001174 DEC WORK
2072 010504 000771 BR 5$
2073 010506 020177 170410 6$: CMP GOOD,@RSAS ;IS RSAS CORRECT?
2074 010512 001403 BEQ 7$ ;YES
2075 010514 017700 170402 MOV @RSAS,BAD ;NO
2076 010520 104000 HLT
2077 010522 010177 170374 7$: MOV GOOD,@RSAS ;CLEAR AS REG
2078 010526 005777 170370 TST @RSAS ;DID IT CLEAR?
2079 010532 001401 BEQ .+4 ;YES
2080 010534 104100 HLT !AS ;NO
2081 010536 022777 010600 170352 CMP #10600,@RSDS ;DID ATA CLEAR?
2082 010544 001401 BEQ .+4 ;YES
2083 010546 104040 HLT !DS ;NO
2084 010550 023777 001102 170334 CMP @#OBUFSV,@RSBA ;DID BA MOVE?
2085 010556 001401 BEQ .+4 ;NO
2086 010560 104021 HLT !CS1!BA ;BA MOVED WHY?
2087 010562 022777 177777 170320 CMP #-1,@RSWC ;DID WC MOVE?
2088 010570 001401 BEQ .+4 ;NO
2089 010572 104010 HLT !WC ;WC MOVED WHY?
2090 010574 000401 BR OOUT ;DENE GET OUT
2091 010576 104040 HLT !DS ;DYR NEVER CAME UP
2092 010600 TTMOUT: OOUT: ;DONE
    
```

```

2093 ;*****
2094 ;TEST 60 ILLEGAL FUNCTION CODE TEST CODE 3 TO 51
2095 ;*****
2096 C10600 104400 TST60: SCOPE
2097
2098 ;TEST ILF BIT IN RSER AND ERR BIT IN RSDS
2099 ;ALSO CHECKS TO SEE IF WC,BA, OR DA GOT MODIFIED
2100 ;IF WISHING TO LOOP ON ONE FUNCTION ONLY, LOAD
2101 ;FUNCTION INTO LOCATION ILLTAB: AND 0 IN FOLLOWING LOCATION
2102
2103 010602 013737 022504 001164 ILL51: MOV TIMES,TIMSV ;SAVE LOOP COUNT
2104 010610 012737 000010 022504 MOV #10,TIMES ;LOOP TEN TIMES
2105 010616 104414 CLRDK ;CLEAR ALL RS REG
2106 010620 012703 027370 1$: MOV #ILLTAB,R3 ;GET STARTING ADD OF TABLE
2107 010624 012300 3$: MOV (R3)+,BAD ;GET ILL FUN
2108 010626 001513 BEQ ILFDN ;DONE GET OUT
2109 010630 013777 001102 170254 MOV @#OBUFSV,@RSBA ;SET UP REGS.
2110 010636 012777 177777 170244 MOV #-1,@RSWC ;TO CHECK FOR CHANGE
2111 010644 010077 170234 2$: MOV BAD,@RSCS1 ;DO ILLEGAL FUNCTION
2112 010650 042700 000001 BIC #BIT0,BAD ;CLEAR GO BIT
2113 010654 010001 MOV BAD,GOOD ;MOV ILLEGAL FUN INTO GOOD
2114 010656 105777 170222 6$: TSTB @RSCS1 ;RDY SET?
2115 010662 100375 BPL 6$ ;NO
2116 010664 052701 104200 BIS #104200,GOOD ;SET ERROR BITS
2117 010670 017700 170210 4$: MOV @RSCS1,BAD ;PUT CS1 INTO BAD
2118 010674 020100 CMP GOOD,BAD ;IS CS1 CORRECT?
2119 010676 001401 BEQ .+4 ;YES
2120 010700 104000 HLT ;GOOD IS WHAT CS1 SHOULD =BAD=CS1
2121 010702 022777 000001 170210 CMP #1,@RSER ;DID ILF SET?
2122 010710 001401 BEQ .+4 ;YES
2123 010712 104043 HLT !CS1!ER!DS ;ILF DID NOT SET
2124 010714 022777 150600 170174 CMP #150600,@RSDS ;IS DS GOOD?
2125 010722 001401 BEQ .+4 ;YES
2126 010724 104043 HLT !CS1!ER!DS ;ERR DID NOT SET
2127 010726 017700 170154 MOV @RSCS2,BAD ;GET CS2
2128 010732 013701 001152 MOV UNNUM,GOOD ;GET UNIT #
2129 010736 052701 000100 BIS #100,GOOD ;SET IR BIT
2130 010742 020100 CMP GOOD,BAD ;IS CS2 CORRECT?
2131 010744 001401 BEQ .+4 ;YES
2132 010746 104000 HLT ;GOOD = CORRECT ANS FOR CS2
2133 010750 013701 001156 MOV UNCMP,GOOD ;GET CORRECT DRIVE
2134 010754 042701 177400 BIC #177400,GOOD ;CLEAR UNWANTED BITS
2135 010760 017700 170136 MOV @RSAS,BAD ;GET RSAS REG
2136 010764 020001 CMP BAD,GOOD ;DID CORRECT UNIT ANSWER?
2137 010766 001401 BEQ .+4 ;YES
2138 010770 104100 HLT !AS ;NG WRONG DRIVE ANSWERED

```

```

2139 010772 023777 001102 170112      CMP      @#0BUF SV,@RSBA      ;DID BA MOVE
2140 011000 001401                      BEQ      .+4                  ;NO
2141 011002 104021                      HLT      !CS1!BA              ;BA MOVED ON AN ILLEGAL FUNCTION
2142 011004 022777 177777 170076      CMP      #-1,@RSWC           ;DID WC MOVE?
2143 011012 001401                      BEQ      .+4                  ;NO
2144 011014 104011                      HLT      !CS1!WC              ;WC MOVED
2145 011016 005777 170072      TST      @RSDA                ;DID DA MOVE
2146 011022 001401                      BEQ      .+4                  ;NO
2147 011024 104005                      HLT      !CS1!DA              ;DA MOVED
2148 011026 104414      CLRDK                                ;CLEAR ALL ERRORS
2149 011030 005777 170064      TST      @RSER                ;DID ERRORS CLEAR
2150 011034 001401                      BEQ      .+4                  ;YES
2151 011036 104040                      HLT      !DS                  ;ILF DID NOT CLEAR
2152 011040 022777 004200 170036      CMP      #4200,@RSCS1        ;DID ERRORS IN CS1 CLEAR
2153 011046 001401                      BEQ      .+4                  ;YES
2154 011050 104040                      HLT      !DS
2155 011052 000137 010624      JMP      3$                    ;CONTINUE UNTIL DONE
2156 011056                      ILFDN:                          ;DONE WITH ILLEGAL FUNCTION TEST
2157                      ;*****
2158                      ;TEST 61                      ILLEGAL FUNCTION CODE TEST CODE 53 TO 77
2159                      ;*****
2160 011056 104400      TST61: SCOPE
2161
2162                      ;TEST ILF BIT IN RSER AND ERR BIT IN RSDS
2163                      ;ALSO CHECKS TO SEE IF WC,BA, OR DA GOT MODIFIED
2164                      ;IF WISHING TO LOOP ON ONE FUNCTION ONLY, LOAD
2165                      ;FUNCTION INTO LOCATION ILFTB2: AND 0 IN FOLLOWING LOCATION
2166
2167 011060 104414      ILLFUN: CLRDK                    ;CLEAR ALL RS REG
2168 011062 012703 027432      1$:  MOV      #ILFTB2,R3        ;GET TABLE OF ILL FUNS.
2169 011066 005037 001174      3$:  CLR      WORK              ;CLEAR WORK
2170 011072 012300                      MOV      (R3)+,BAD            ;GET ILL FUN
2171 011074 001554                      BEQ      ILFDNE               ;DONE GET OUT
2172 011076 013777 001102 170006      MOV      @#0BUF SV,@RSBA     ;SET UP REGS.
2173 011104 012777 177777 167776      MOV      #-1,@RSWC           ;TO CHECK FOR CHANGE
2174 011112 010037 001176                      MOV      BAD,WORK1           ;SHOULD WE TEST
2175 011116 042737 177707 001176      BIC      #177707,WORK1       ;BA AND WC
2176 011124 022737 000060 001176      CMP      #60,WORK1           ;TO INC
2177 011132 001003                      BNE      2$                   ;NO
2178 011134 012737 000007 001174      MOV      #7,WORK             ;YES
2179 011142 010077 167736      2$:  MOV      BAD,@RSCS1        ;DO ILLEGAL FUNCTION
2180 011146 042700 000001      10$: BIC      #BIT0,BAD          ;CLEAR GO BIT
2181 011152 010001                      MOV      BAD,GOOD            ;MOV ILLEGAL FUN INTO GOOD
    
```

```

2182 011154 105777 167724      6$:  TSTB  @RSCS1      ;RDY SET?
2183 011160 100375                BPL    6$          ;NO
2184 011162 052701 144200        BIS    #144200,GOOD ;SET ERROR BITS
2185 011166 017700 167712        MOV    @RSCS1,BAD  ;PUT CS1 INTO BAD
2186 011172 020100                CMP    GOOD,BAD    ;IS CS1 CORRECT?
2187 011174 001401                BEQ    .+4         ;YES
2188 011176 104000                HLT                    ;GOOD IS WHAT CS1 SHOULD =BAD=CS1
2189 011200 022777 000001 167712    CMP    #1,@RSER    ;DID ILF SET?
2190 011206 001401                BEQ    .+4         ;YES
2191 011210 104043                HLT    !CS1!ER!DS  ;ILF DID NOT SET
2192 011212 022777 150600 167676    CMP    #150600,@RSDS ;IS DS GOOD?
2193 011220 001401                BEQ    .+4         ;YES
2194 011222 104043                HLT    !CS1!ER!DS  ;ERR DID NOT SET
2195 011224 005777 167664        TST    @RSDA       ;DID DA MOVE?
2196 011230 001401                BEQ    .+4         ;NO
2197 011232 104005                HLT    !CS1!DA     ;DA MOVED
2198 011234 005737 001174        TST    WORK        ;IS THIS AN ILL WRITE FUN?
2199 011240 001025                BNE    11$        ;YES
2200 011242 017700 167640        MOV    @RSCS2,BAD  ;GET CS2
2201 011246 013701 001152        MOV    UNNUM,GOOD  ;GET UNIT #
2202 011252 052701 001100        BIS    #1100,GOOD  ;SET IR BIT
2203 011256 020100                CMP    GOOD,BAD    ;IS CS2 CORRECT?
2204 011260 001401                BEQ    .+4         ;YES
2205 011262 104000                HLT                    ;GOOD = CORRECT ANS FOR CS2
2206 011264 023777 001102 167620    CMP    @#OBUFSV,@RSBA ;DID BA MOVE
2207 011272 001401                BEQ    .+4         ;NO
2208 011274 104021                HLT    !CS1!BA     ;BA MOVED ON AN ILLEGAL FUNCTION
2209 011276 022777 177777 167604    CMP    #-1,@RSWC   ;DID WC MOVE?
2210 011304 001401                BEQ    .+4         ;NO
2211 011306 104011                HLT    !CS1!WC     ;WC MOVED
2212 011310 000137 011376                JMP    4$          ;CONTINUE UNTIL DONE
2213 011314 017700 167566                MOV    @RSCS2,BAD  ;GET CS2
2214 011320 013701 001152                MOV    UNNUM,GOOD  ;GET UNIT #
2215 011324 052701 001300                BIS    #1300,GOOD  ;SET IR BIT
2216 011330 020100                CMP    GOOD,BAD    ;IS CS2 CORRECT?
2217 011332 001401                BEQ    .+4         ;YES
2218 011334 104000                HLT                    ;GOOD = CORRECT ANS FOR CS2
2219 011336 013737 001102 001174    MOV    @#OBUFSV,WORK ;GET BUFFER ADDR.
2220 011344 062737 000002 001174    ADD    #2,WORK     ;UPDATE IT
2221 011352 023777 001174 167532    CMP    WORK,@RSBA  ;DID BA MOVE
2222 011360 001401                BEQ    .+4         ;YES
2223 011362 104021                HLT    !CS1!BA     ;BA MOVED ON AN ILLEGAL FUNCTION
2224 011364 022777 000000 167516    CMP    #0,@RSWC   ;DID WC MOVE?
2225 011372 001401                BEQ    .+4         ;NO
2226 011374 104011                HLT    !CS1!WC     ;WC MOVED
2227 011376 104414                CLRDK                ;CLEAR ALL ERRORS
2228 011400 022777 004200 167476    CMP    #4200,@RSCS1 ;DID ERRORS CLEAR
2229 011406 001401                BEQ    .+4         ;YES
2230 011410 104040                HLT    !DS         ;NO
2231 011412 005777 167502        TST    @RSER       ;DID ERROR CLEAR
2232 011416 001401                BEQ    .+4         ;NO
2233 011420 104040                HLT    !DS         ;YES
2234 011422 000137 011066                JMP    3$          ;CONTINUE UNTIL DONE
2235 011426                ILFDNE:           ;DONE WITH ILLEGAL FUNCTION TEST
    
```



```

2236 :*****
2237 :TEST 62 TEST ILLEGAL FUNCTION CODE 67
2238 :*****
2239 011426 104400 TST62: SCOPE
2240
2241 011430 104414 ILF67: CLRDK :CLEAR ALL RS REG
2242 011432 012777 177777 167450 MOV #-1,@RSWC :SET WC TO -1
2243 011440 013737 001102 001174 MOV @#OBUFSV,WORK :GET OUTBUF ADD.
2244 011446 062737 000002 001174 ADD #2,WORK :FOR TEST
2245 011454 013777 001174 167430 MOV WORK,@RSBA :LOAD ADDR.
2246 011462 012777 000067 167414 MOV #67,@RSCS1 :DO FUNCTION 67
2247 011470 105777 167410 1$: TSTB @RSCS1 :DONE YET?
2248 011474 100375 BPL 1$ :NO
2249 011476 017700 167410 MOV @RSBA,BAD :GET BA REG
2250 011502 013701 001102 MOV @#OBUFSV,GOOD :GET CORRECT ANS FOR RSBA
2251 011506 020100 CMP GOOD,BAD :IS RSBA CORRECT?
2252 011510 001401 BEQ .+4 :YES
2253 011512 104000 HLT :BAD=RSBA GOOD=CORRECT ANS.
2254 011514 013701 001152 MOV UNNUM,GOOD :GET UNIT NUMBER
2255 011520 052701 001300 BIS #1300,GOOD :SET IR AND OR BITS
2256 011524 017700 167356 MOV @RSCS2,BAD :GET CS2
2257 011530 020100 CMP GOOD,BAD :IS CS2 CORRECT?
2258 011532 001401 BEQ .+4 :YES
2259 011534 104000 HLT :BAD=CS2 GOOD=CORRECT ANS
2260 011536 022777 000001 167354 CMP #1,@RSER :IS RSER CORRECT?
2261 011544 001401 BEQ .+4 :YES
2262 011546 104002 HLT !ER :ER IS WRONG
2263 011550 104414 CLRDK :CLEAR ALL RS REG
2264 011552 005777 167342 TST @RSER :DID ERROR CLEAR
2265 011556 001401 BEQ .+4 :YES
2266 011560 104040 HLT !DS :NO
2267 011562 012777 177700 167320 MOV #-100,@RSWC :SET WC TO -100
2268 011570 013737 001102 001174 MOV @#OBUFSV,WORK :GET OUTBUF ADD.
2269 011576 062737 000200 001174 ADD #200,WORK :FOR TEST
2270 011604 013777 001174 167300 MOV WORK,@RSBA :LOAD ADDR
2271 011612 012777 000067 167264 MOV #67,@RSCS1 :DO FUNCTION 67
2272 011620 105777 167260 2$: TSTB @RSCS1 :DONE YET?
2273 011624 100375 BPL 2$ :NO
    
```

```

2274 011626 013701 001102      3$:  MOV @#OBUFSV,GOOD ;GET CORRECT ANS.
2275 011632 017700 167254      MOV @RSBA,BAD ;GET BA REG
2276 011636 020100      CMP GOOD,BAD ;IS RSBA CORRECT?
2277 011640 001401      BEQ .+4 ;YES
2278 011642 104000      HLT ;BAD=RSBA GOOD=CORRECT ANS.
2279 011644 013701 001152      MOV UNNUM,GOOD ;GET UNIT NUMBER
2280 011650 052701 001300      BIS #1300,GOOD ;SET IR AND OR BITS
2281 011654 017700 167226      MOV @RSCS2,BAD ;GET CS2
2282 011660 020100      CMP GOOD,BAD ;IS CS2 CORRECT?
2283 011662 001401      BEQ .+4 ;YES
2284 011664 104000      HLT ;BAD=CS2 GOOD=CORRECT ANS
2285 011666 022777 000001 167224      CMP #1,@RSER ;IS RSER CORRECT?
2286 011674 001401      BEQ .+4 ;YES
2287 011676 104002      HLT !ER ;ER IS WRONG
2288 011700 012777 040011 167176      MOV #40011,@RSCS1 ;CLEAR ERRORS
2289 011706 022777 004210 167170      CMP #4210,@RSCS1 ;DID THEY CLEAR IN CS1
2290 011714 001401      BEQ .+4 ;YES
2291 011716 104040      HLT !DS ;NO
2292 011720 005777 167174      TST @RSER ;DID RSER CLEAR
2293 011724 001401      BEQ .+4 ;YES
2294 011726 104040      HLT !DS ;NO
2295 011730 005777 167166      TST @RSAS ;DID RSAS CLEAR
2296 011734 001401      BEQ .+4 ;YES
2297 011736 104100      HLT !AS ;NO

```

```

2298
2299
2300 ;*****
2301 ;TEST 63 TEST PAR IN RSER
2302 ;*****

```

```

2302 011740 104400      TST63: SCOPE
2303
2304 011742 013737 001164 022504      PARTST: MOV TIMSV,TIMES ;RESTORE LOOP #
2305 011750 104414      CLRDK ;CLEAR ALL RS REG
2306 011752 012777 000010 167140      1$: MOV #10,@RSER ;SET PAR
2307 011760 022777 150600 167130      CMP #150600,@RSDS ;DID ERR,ATA AND DRY SET?
2308 011766 001401      BEQ .+4 ;YES
2309 011770 104042      HLT !DS!ER ;ER SHOULD SET IF PAR SETS IN RSER
2310 011772 104414      CLRDK ;CLEAR ALL RS REG
2311 011774 005777 167120      TST @RSER ;DID PAR CLEAR?
2312 012000 001401      BEQ .+4 ;YES
2313 012002 104002      HLT !ER ;PAR DID NOT CLEAR BY CLEAR BIT
2314 012004 022777 010600 167104      CMP #10600,@RSDS ;DID ERROR BITS CLEAR
2315 012012 001401      BEQ .+4 ;YES
2316 012014 104040      HLT !DS ;NO

```

```

2317 ;CHECK BITS 12 TO 15 FOR 0
2318 ;CHECK SECTOR FRACTION TO WATCH FOR MOVEMENT
2319 ;CHECK CS BITS IN LA AND ADDRESS CONFIRM IN MR REG
2320
2321 ;*****
2322 ;TEST 64 LOOK AHEAD TEST
2323 ;*****
2324 012016 104400 TST64: SCOPE
2325
2326 012020 032777 170000 167076 LATST: BIT #170000,@RSLA ;ARE BITS 12 TO 15 CLEARED?
2327 012026 001401 BEQ .+4 ;YES
2328 012030 104204 HLT !LA ;BITS 12 TO 15 SHOULD BE CLEARED
2329 012032 104400 TST65: SCOPE
2330
2331 ;NOW TEST MOVEMENT IN SF BITS
2332
2333 012034 012737 171005 001174 MOV #171005,WORK ;SET UP FOR TIME OUT ROUTINE
2334 012042 017701 167056 MOV @RSLA,GOOD ;GET READING FROM LA
2335 012046 042701 007700 BIC #7700,GOOD ;GET RID OF CS BITS
2336 012052 005337 001174 1$: DEC WORK ;WAIT FOR DISK
2337 012056 001407 BEQ ERRR ;TYPE ERROR
2338 012060 017700 167040 MOV @RSLA,BAD ;READ LA
2339 012064 042700 007700 BIC #7700,BAD ;CLEAR CS BITS
2340 012070 020100 CMP GOOD,BAD ;DID SF BITS CHANGE?
2341 012072 001767 BEQ 1$ ;WAIT FOR TIME OUT
2342 012074 000422 BR LATDON ;LA OK CONT
2343 012076 ERRR:
2344 012076 104402 012102 TYPE ..+2 ;.ASCIZ <15><12>'SECTOR FRACTIONS NOT MOVING'
2345 012140 104204 HLT !LA ;TYPE LOOK AHEAD REG
2346 012142 LATDON: ;DONE CONT.
    
```

```

2347
2348
2349
2350 012142 104400
2351
2352 012144 013737 022504 001164 CSTST: MOV TIMES,TIMSV ;SAVE LOOP CT
2353 012152 012737 000010 022504 MOV #10,TIMES ;LOOP 10 TIMES
2354 012160 104414 CLRDK ;CLEAR ALL RS REG.
2355 012162 012701 001000 MOV #1000,GOOD ;LOAD COUNTER
2356 012166 032777 001000 166734 BIT #BIT9,@RSMR ;IS ADD CONFIRM BIT 0?
2357 012174 001407 BEQ ADDCF ;YES CONTINUE
2358 012176 005301 DEC GOOD ;WAIT FOR
2359 012200 001376 BNE -2 ;DISK TO MOVE
2360 012202 032777 001000 166720 BIT #BIT9,@RSMR ;IS AADD. CON. BIT BIT 0?
2361 012210 001401 BEQ +4 ;YES
2362 012212 104220 HLT !MR ;ADD. CONF. BIT ALWAYS A 1
2363
2364 ;NOW TEST TA BITS AND ADD. CON. BIT IN MR
2365
2366 012214 012777 177777 166672 ADDCF: MOV #-1,@RSDA ;INIT RSDA
2367 012222 012737 177777 001174 1$: MOV #-1,WORK ;SETUP TIMEOUT COUNTER
2368 012230 005277 166660 INC @RSDA ;GET NEXT SECTOR
2369 012234 022777 010000 166652 CMP #10000,@RSDA ;DONE ALL YET?
2370 012242 001433 BEQ DONCS ;YES
2371 012244 005337 001174 2$: DEC WORK ;DO TIMEOUT ROUTINE
2372 012250 001427 BEQ TMEOUT ;ADD. CON. NEVER CAME UP
2373 012252 032777 001000 166650 BIT #1000,@RSMR ;DID ADD CONFIRM BIT SET?
2374 012260 001771 BEQ 2$ ;YES
2375 012262 017700 166636 MOV @RSLA,BAD ;GET LA
2376 012266 042700 000077 BIC #77,BAD ;CLEAR SF BITS
2377 012272 012737 000006 001176 MOV #6,WORK1 ;SET UP COUNTER
2378 012300 006000 ROR BAD ;MOV SA BITS RIGHT
2379 012302 005337 001176 3$: DEC WORK1 ;DO 6 TIMES
2380 012306 001374 BNE 3$ ;DONE YET?
2381 012310 017701 166600 MOV @RSDA,GOOD ;GET DA
2382 012314 042701 177700 BIC #177700,GOOD ;CLEAR JUNK
2383 012320 020100 CMP GOOD,BAD ;ARE SA BITS = IN DA AND LA REG.?
2384 012322 001401 BEQ +4 ;OK
2385 012324 104000 HLT ;GOOD =DA BAD = LA
2386 012326 000735 BR 1$ ;NO WAIT
2387
2388 012330 104262 TMEOUT: HLT !MR!ER!DS ;ADDRESS CONFIRM BIT NEVER SET COULD BE BAD OR
2389
2390 012332 DONCS: ;BAD LA OR BAD COMPARE BETWEEN LA AND DA
;TEST DONE CONTINUE
    
```

```

2391 :*****
2392 :TEST 67 PARITY TEST
2393 :*****
2394 012332 104400 TST67: SCOPE
2395
2396 012334 012701 031054 PART: MOV #PARITY,R1 ;GET TEST LOCATION
2397 012340 013737 001164 022504 MOV TIMSV,TIMES ;RESTORE LOOP COUNTER
2398 012346 012737 012512 000004 MOV #PRTP,@#4 ;SETUP TIME OUT VECTOR
2399 012354 012737 000340 000006 MOV #340,@#6
2400 012362 012702 172100 MOV #MPRO,R2 ;GET PAR REG
2401 012366 005712 TSTAGN: TST (R2) ;DOES IT EXIST
2402 012370 012712 000004 MOV #WWP,@R2 ;YES SET WRITE WRONG PARITY
2403 012374 011111 1$: MOV @R1,@R1 ;WRITE WRONG PARITY
2404 012376 005711 TST @R1 ;READ IT
2405 012400 005712 TST @R2 ;DID PARITY ERROR SET?
2406 012402 100402 BMI 2$ ;YES
2407 012404 005012 CLR @R2 ;CLEAR PARITY REG
2408 012406 000442 BR PRTP1 ;GET NEXT PARITY REG
2409 012410 042712 100005 2$: BIC #100005,@R2 ;TURN OFF WWP,ENABLE BAD PARITY, ACTION ENABLE
2410 012414 005712 TST @R2 ;PARITY ERROR?
2411 012416 001445 BEQ NOPAR ;IF NO BR.
2412
2413 012420 104414 CLRDK ;CLEAR ALL RS REG
2414 012422 012777 031054 166462 MOV #PARITY,@RSBA ;SET UP CURRENT ADDRESS
2415 012430 012777 177777 166452 MOV #-1,@RSWC ;SET WORD COUNT TO -1
2416 012436 012777 000061 166440 MOV #61,@RSCS1 ;GO WRITE
2417 012444 105777 166434 3$: TSTB @RSCS1 ;DONE YET?
2418 012450 100375 BPL 3$ ;NO WAIT
2419 012452 017700 166430 MOV @RSCS2,BAD ;GET CS2
2420 012456 012701 020100 MOV #20100,GOOD
2421 012462 053701 001152 BIS UNNUM,GOOD ;GET CORRECT AND FOR CS2
2422 012466 020100 CMP GOOD,BAD ;IS CS2 CORRECT?
2423 012470 001401 BEQ .+4 ;YES
2424 012472 104000 HLT ;CS2 SHOULD = GOOD
2425 012474 022777 144260 166402 CMP #144260,@RSCS1 ;IS CS1 CORRECT?
2426 012502 001401 BEQ .+4 ;YES
2427 012504 104040 HLT !DS
2428 012506 000137 012532 JMP NOPAR ;GET OUT
2429
2430 ;TRAPOUT ROUTINE
2431
2432 012512 022626 PRTP: CMP (6)+,(6)+ ;CLEAR STACK
2433 012514 022702 172136 PRTP1: CMP #172136,R2 ;DONE YET?
2434 012520 001404 BEQ NOPAR ;YES NO PAR REG
2435 012522 062702 000002 ADD #2,R2 ;NO TRY AGAIN
2436 012526 000137 012366 JMP TSTAGN ;RETRY
2437
2438 012532 012737 000006 000004 NOPAR: MOV #6,@#4
2439 012540 005037 000006 CLR @#C
2440 012544 005037 031054 CLR PARITY ;WRITE GOOD PARITY
2441 :*****
2442 :TEST 70 TEST WRITE CHECK ERROR
2443 :*****
2444 012550 104400 TST70: SCOPE
2445
2446 ;WRITE A WORD OF 0 AND FLOAT A 1 THROUGH IT TO CAUSE WCE
    
```

```

2447 ;SET BIT14 IN ONCEE AND WRITE A WD OF -1 AND FLOAT 0
2448 ;TO CAUSE WCE
2449
2450 012552 104414 WCETST: CLRDK ;CLEAR ALL RS REG
2451 012554 005037 027454 CLR OUTBUF ;WRITE A WD OF 0
2452 012560 013777 001102 166324 WCETT: MOV @#OBUF SV,@RSBA ;SET UP CURRENT ADDRESS
2453 012566 012777 177777 166314 MOV #-1,@RSWC ;SET WORD COUNT TO -1
2454 012574 012777 000061 166302 MOV #61,@RSCS1 ;GO WRITE
2455 012602 105777 166276 3$: TSTB @RSCS1 ;DONE YET?
2456 012606 100375 BPL 3$ ;NO WAIT
2457 012610 032737 040000 001160 BIT #BIT14,ONCEE ;WRITE A 1 OR 0?
2458 012616 001410 BEQ 2$ ;WRITE A 0
2459 012620 012737 177777 027454 MOV #-1,OUTBUF ;WRITE A 1
2460 012626 000241 CLC ;CLEAR CARRY
2461 012630 006137 027454 6$: ROL OUTBUF ;FLOAT A 0 THROUGH BAD WD
2462 012634 103123 BCC WCEDON ;DONE GET OUT
2463 012636 000406 BR 5$ ;CHECK WCE
2464 012640 005037 027454 2$: CLR OUTBUF ;WRITE A 0
2465 012644 000261 SEC ;SET CARRY
2466 012646 006137 027454 1$: ROL OUTBUF ;FLOAT A 1
2467 012652 103503 BCS WCEDNE ;GET OUT WHEN DONE
2468 012654 013777 001102 166230 5$: MOV @#OBUF SV,@RSBA ;SET UP CURRENT ADDRESS
2469 012662 012777 177777 166220 MOV #-1,@RSWC ;SET WORD COUNT TO -1
2470 012670 005077 166220 CLR @RSDA
2471 012674 012777 000051 166202 MOV #51,@RSCS1 ;GO WRITE CHECK
2472 012702 105777 166176 4$: TSTB @RSCS1 ;READY YET?
2473 012706 100375 BPL 4$ ;NO WAIT
2474 012710 017700 166172 MOV @RSCS2,BAD ;GET CS2
2475 012714 013701 001152 MOV UNNUM,GOOD ;SET UNIT #
2476 012720 052701 040300 BIS #40300,GOOD ;SET BITS
2477 012724 020100 CMP GOOD,BAD ;IS CS2 CORRECT?
2478 012726 001413 BEQ 7$ ;YES
2479 012730 104000 HLT ;BAD=CS2 GOOD=CORRECT ANS
2480 012732 013700 027454 MOV OUTBUF,BAD ;GET BAD WD THAT SHOULD CAUSE WCE
2481 012736 005001 CLR GOOD ;GET GOOD WD IF WRITING 0
2482 012740 032737 040000 001160 BIT #BIT14,ONCEE ;ARE WE WRITING 1 OR 0
2483 012746 001402 BEQ 8$ ;0
2484 012750 012701 177777 MOV #-1,GOOD ;GET GOOD WD FOR 1
2485 012754 104000 8$: HLT ;GOOD = CORRECT WD WRITTEN
2486 ;BAD = INCORRECT WD THAT WCE DID NOT CATCH
  
```

2487	012756	022777	144250	166120	7\$:	CMP	#144250,@RSCS1	:DID TRE SET?
2488	012764	001401				BEQ	+.4	:YES
2489	012766	104043				HLT	!CS1!ER!DS	:TRE SHOULD SET IF WCE SETS
2490	012770	017700	166116			MOV	@RSBA,BAD	:FETCH CURRENT ADDRESS
2491	012774	013701	001102			MOV	@#OBUFSV,GOOD	:WHAT RSBA SHOULD EQUAL
2492	013000	062701	000002			ADD	#2,GOOD	:UPDATE IT
2493	013004	020001				CMP	BAD,GOOD	:IS RSBA CORRECT
2494	013006	001401				BEQ	+.4	:YES EXECUTE CONTINUE
2495	013010	104000				HLT		:RSBA FAILED TO INCREMENT
2496	013012	104414				CLRDK		:CLEAR ALL RS REG
2497	013014	013701	001152			MOV	UNNUM,GOOD	:PUT DRIVE IN GOOD
2498	013020	052701	000100			BIS	#100,GOOD	:SET IR BIT
2499	013024	017700	166056			MOV	@RSCS2,BAD	:GET CS2
2500	013030	020100				CMP	GOOD,BAD	:IS CS2 CORRECT
2501	013032	001401				BEQ	+.4	:YES
2502	013034	104000				HLT		:BAD =CS2 GOOD IS CORRECT ANS
2503	013036	022777	004200	166040		CMP	#4200,@RSCS1	:DID TRE CLEAR?
2504	013044	001401				BEQ	+.4	:YES
2505	013046	104001				HLT	!CS1	:TRE DID NOT CLEAR WITH CLEAR
2506	013050	032737	040000	001160		BIT	#BIT14,ONCEE	:FLOATION A 1 OR 0?
2507	013056	001673				BEQ	1\$	:FLOAT 1
2508	013060	000663				BR	6\$	:FLOAT 0
2509	013062	052737	040000	001160	WCEDNE:	BIS	#BIT14,ONCEE	:SET BIT14
2510	013070	104414				CLRDK		
2511	013072	012737	177777	027454		MOV	#-1,OUTBUFF	
2512	013100	000137	012560			JMP	WCETT	:NOW WRITE -1 IN OUTBUF
2513	013104	042737	040000	001160	WCEDON:	BIC	#BIT14,ONCEE	:CLEAR TEST FLAG

```

2514 :*****
2515 :TEST 71 TEST WRITE CHECK ERROR ON -B- PORT
2516 :*****
2517 013112 104400 TST71: SCOPE
2518
2519 :WRITE A WORD OF 0 AND FLOAT A 1 TRSOUGH IT TO CAUSE WCE
2520 :SET BIT14 IN ONCEE AND WRITE A WD OF -1 AND FLOAT 0
2521 :TO CAUSE WCE
2522
2523 013114 032737 020000 001160 WCETSB: BIT #BIT13,ONCEE :-B- PORT?
2524 013122 001402 BEQ 1$ :YES
2525 013124 000137 013470 JMP WCEDOS :NO GET OUT
2526 013130 013737 001170 001102 1$: MOV BPORTT,OBUSV :GET -B- PORT BUFFER
2527 013136 104414 CLRDK :CLEAR ALL RS REG
2528 013140 005077 166024 CLR @BPORTT :WRITE A WD OF 0
2529 013144 013777 001102 165740 WCETB: MOV @#OBUSV,@RSBA :SET UP CURRENT ADDRESS
2530 013152 012777 177777 165730 MOV #-1,@RSWC :SET WORD COUNT TO -1
2531 013160 012777 002061 165716 MOV #2061,@RSCS1 :GO WRITE
2532 013166 105777 165712 3$: TSTB @RSCS1 :DONE YET?
2533 013172 100375 BPL 3$ :NO WAIT
2534 013174 032737 040000 001160 BIT #BIT14,ONCEE :WRITE A 1 OR 0?
2535 013202 001410 BEQ 2$ :WRITE A 0
2536 013204 012777 177777 165756 MOV #-1,@BPORTT :WRITE A 1
2537 013212 000241 CLC :CLEAR CARRY
2538 013214 006137 027454 6$: ROL OUTBUF :FLOAT A 0 THROUGH BAD WD
2539 013220 103123 BCC WCEDOS :DONE GET OUT
2540 013222 000406 BR 5$ :CHECK WCE
2541 013224 005077 165740 2$: CLR @BPORTT :WRITE A 0
2542 013230 000261 SEC :SET CARRY
2543 013232 006177 165732 1$: ROL @BPORTT :FLOAT A 1
2544 013236 103503 BCS WCEDNB :GET OUT WHEN DONE
2545 013240 013777 001102 165644 5$: MOV @#OBUSV,@RSBA :SET UP CURRENT ADDRESS
2546 013246 012777 177777 165634 MOV #-1,@RSWC :SET WORD COUNT TO -1
2547 013254 005077 165634 CLR @RSDA
2548 013260 012777 002051 165616 MOV #2051,@RSCS1 :GO WRITE CHECK
2549 013266 105777 165612 4$: TSTB @RSCS1 :READY YET?
2550 013272 100375 BPL 4$ :NO WAIT
2551 013274 017700 165606 MOV @RSCS2,BAD :GET CS2
2552 013300 013701 001152 MOV UNNUM,GOOD :SET UNIT #
2553 013304 052701 040300 BIS #40300,GOOD :SET BITS
2554 013310 020100 CMP GOOD,BAD :IS CS2 CORRECT?
2555 013312 001413 BEQ 7$ :YES
2556 013314 104000 HLT :CS2=BAD GOOD=CORRECT ANS
2557 013316 017700 165646 MOV @BPORTT,BAD :GET BAD WD THAT SHOULD CAUSE WCE
2558 013322 005001 CLR GOOD :GET GOOD WD IF WRITING 0
2559 013324 032737 040000 001160 BIT #BIT14,ONCEE :ARE WE WRITING 1 OR 0
2560 013332 001402 BEQ 8$ :0
2561 013334 012701 177777 MOV #-1,GOOD :GET GOOD WD FOR 1
2562 013340 104000 8$: HLT :GOOD = CORRECT WD WRITTEN
2563 :BAD = INCORRECT WD THAT WCE DID NOT CATCH
    
```



```

2564 013342 022777 146250 165534 7$:  CMP      #146250,@RSCS1  :DID TRE SET?
2565 013350 001401  BEQ      .+4      :YES
2566 013352 104043  HLT      !CS1!ER!DS :TRE SHOULD SET IF WCE SETS
2567 013354 017700 165532  MOV      @RSBA,BAD   :FETCH CURRENT ADDRESS
2568 013360 013701 001102  MOV      @#OBUFSV,GOOD :WHAT RSBA SHOULD EQUAL
2569 013364 062701 000002  ADD      #2,GOOD    :UPDATE IT
2570 013370 020001  CMP      BAD,GOOD   :IS RSBA CORRECT
2571 013372 001401  BEQ      .+4      :YES EXECUTE CONTINUE
2572 013374 104000  HLT      :RSBA FAILED TO INCREMENT
2573 013376 104414  CLRDK    :CLEAR ALL RS REG
2574 013400 013701 001152  MOV      UNNUM,GOOD  :PUT DRIVE IN GOOD
2575 013404 052701 000100  BIS      #100,GOOD   :SET IR BIT
2576 013410 017700 165472  MOV      @RSCS2,BAD  :GET CS2
2577 013414 020100  CMP      GOOD,BAD   :IS CS2 CORRECT
2578 013416 001401  BEQ      .+4      :YES
2579 013420 104000  HLT      :BAD =CS2 GOOD IS CORRECT ANS
2580 013422 022777 004200 165454  CMP      #4200,@RSCS1 :DID TRE CLEAR?
2581 013430 001401  BEQ      .+4      :YES
2582 013432 104001  HLT      !CS1      :TRE DID NOT CLEAR WITH CLEAR
2583 013434 032737 040000 001160  BIT      #BIT14,ONCEE :FLOATION A 1 OR 0?
2584 013442 001673  BEQ      1$       :FLOAT 1
2585 013444 000663  BR       6$       :FLOAT 0
2586 013446 052737 040000 001160  WCEDNB: BIS      #BIT14,ONCEE :SET BIT14
2587 013454 104414  CLRDK
2588 013456 012777 177777 165504  MOV      #-1,@BPORTT
2589 013464 000137 013144  JMP      WCETB     :NOW WRITE -1 IN OUTBUF
2590 013470 012737 027454 001102  WCEDOS: MOV      #OUTBUF,OBUFSV :RESTORE OBUFSV
2591 013476 042737 040000 001160  BIC      #BIT14,ONCEE :CLEAR TEST FLAG
  
```



```

2634
2635
2636
2637 013704 104400
2638
2639 013706 104414
2640 013710 013777 001102 165174
2641 013716 012777 177700 165164
2642 013724 012703 172060
2643 013730 011304
2644 013732 042704 000017
2645 013736 022704 000020
2646 013742 001372
2647 013744 012777 000031 165132
2648 013752 012777 007777 165134
2649 013760 004737 026670
2650 013764 104001
2651 013766 022777 000004 165124
2652 013774 001401
2653 013776 104002
2654 014000 022777 007777 165106
2655 014006 001001
2656 014010 104004
2657 014012 022777 150600 165076
2658 014020 001401
2659 014022 104042
2660 014024 022777 104230 165052
2661 014032 001401
2662 014034 104040
2663 014036 104414
2664 014040 005777 165054
2665 014044 001401
2666 014046 104002
2667 014050 022777 004200 165026
2668 014056 001401
2669 014060 104040
2670
2671
2672
2673
2674 014062 104400
2675
2676 014064 104414
2677 014066 013777 001102 165016
2678 014074 012777 177700 165006
2679 014102 012777 000061 164774
2680 014110 105777 164770
2681 014114 100775
2682 014116 012777 177773 164774
    
```

```

:*****
:TEST 73 TEST RMR IN RSER REGISTER TRYING TO WRITE INTO RSDA
:*****
TST73: SCOPE
    
```

```

RMRT1: CLRDK ;CLEAR ALL RS REG
MOV @#OBUFSV,@RSBA ;SET UP CURRENT ADDRESS
MOV #177700,@RSWC ;SET WORD COUNT
MOV #172060,R3 ;GET RSLA REG
1$: MOV @R3,R4 ;WAIT FOR
BIC #17,R4 ;THE MIDDLE
CMP #20,R4 ;OF SECTOR 0
BNE 1$ ;BEFORE DOING A SEARCH
MOV #31,@RSCS1 ;SEARCH
MOV #7777,@RSDA ;CAUSE ERROR
JSR PC,WAITRY ;WAIT FOR READY
HLT !CS1 ;RDY NEVER CAME UP
CMP #4,@RSER ;DID RMR SET?
BEQ .+4 ;YES
HLT !ER ;ER SHOULD = 4
CMP #7777,@RSDA ;DID DA GET MODIFIED?
BNE .+4 ;NO
HLT !DA
CMP #150600,@RSDS ;DID ERR SET?
BEQ .+4 ;YES
HLT !DS!ER ;ER DID NOT SET BECAUSE OF RMR
CMP #104230,@RSCS1 ;IS CS1 CORRECT?
BEQ .+4 ;YES
HLT !DS ;CS1 SHOULD = 144260
CLRDK ;CLEAR ALL RS REG
TST @RSER ;DID RMR CLEAR?
BEQ .+4 ;YES
HLT !ER ;RMR DID NOT CLEAR WITH A CLEAR
CMP #4200,@RSCS1 ;IS CS1 CORRECT?
BEQ .+4 ;YES
HLT !DS ;NO
    
```

```

:*****
:TEST 74 TEST RMR IN RSER REGISTER TRYING TO WRITE INTO RSER
:*****
TST74: SCOPE
    
```

```

RMRT2: CLRDK ;CLEAR ALL RS REG
MOV @#OBUFSV,@RSBA ;SET UP CURRENT ADDRESS
MOV #177700,@RSWC ;SET WORD COUNT
MOV #61,@RSCS1 ;GC WRITE
2$: TSTB @RSCS1 ;IS RDY SET?
BMI 2$ ;YES WAIT FOR IT TO CLEAR
MOV #177773,@RSER ;CAUSE ERROR
    
```

```

2683 014124 004737 026670 JSR PC, WAITRY
2684 014130 104001 HLT !CS1 ;RDY NEVER CAME UP
2685 014132 022777 000004 164760 CMP #4, @RSER ;DID RMR SET?
2686 014140 001401 BEQ .+4 ;YES
2687 014142 104002 HLT !ER ;ER SHOULD = 4
2688 014144 022777 150600 164744 CMP #150600, @RSDS ;DID ERR SET?
2689 014152 001401 BEQ .+4 ;YES
2690 014154 104042 HLT !DS!ER ;ERR DID NOT SET BECAUSE OF RMR
2691 014156 022777 144260 164720 CMP #144260, @RSCS1 ;IS CS1 CORRECT?
2692 014164 001401 BEQ .+4 ;YES
2693 014166 104040 HLT !DS ;CS1 SHOULD = 144260
2694 014170 104414 CLRDK ;CLEAR ALL RS REG
2695
2696
2697
2698

```

```

:*****
:TEST 75 TEST RMR IN RSER REGISTER TRYING TO WRITE INTO RSCS1
:*****
TST75: SCOPE

```

```

2699 014172 104400
2700
2701 014174 104414 RMRT3: CLRDK ;CLEAR ALL RS REG
2702 014176 013777 001102 164706 MOV @#OBUFSV, @RSBA ;SET UP CURRENT ADDRESS
2703 014204 012777 177700 164676 MOV #177700, @RSWC ;SET WORD COUNT
2704 014212 012777 000061 164664 MOV #61, @RSCS1 ;GO WRITE
2705 014220 105777 164660 2$: TSTB @RSCS1 ;IS RDY SET?
2706 014224 100775 BMI 2$ ;YES WAIT FOR IT TO CLEAR
2707 014226 012777 000030 164650 MOV #30, @RSCS1 ;CAUSE ERROR
2708 014234 004737 026670 JSR PC, WAITRY ;WAIT FOR READY
2709 014240 104001 HLT !CS1 ;RDY NEVER CAME UP
2710 014242 022777 000004 164650 CMP #4, @RSER ;DID RMR SET?
2711 014250 001401 BEQ .+4 ;YES
2712 014252 104002 HLT !ER ;ER SHOULD = 4
2713 014254 022777 150600 164634 CMP #150600, @RSDS ;DID ERR SET?
2714 014262 001401 BEQ .+4 ;YES
2715 014264 104042 HLT !DS!ER ;ERID NOT SET BECAUSE OF RMR
2716 014266 022777 144260 164610 CMP #144260, @RSCS1 ;IS CS1 CORRECT?
2717 014274 001401 BEQ .+4 ;YES
2718 014276 104040 HLT !DS ;CS1 SHOULD = 144260
2719 014300 104414 CLRDK ;CLEAR ALL RS REG

```

```

2720 :*****
2721 :TEST 76 TEST THAT RMR DOES NOT SET BY WRITTING INTO RSAS
2722 :*****
2723 014302 104400 TST76: SCOPE
2724
2725 014304 104414 RMRT4: CLRDK ;CLEAR ALL RS REG
2726 014306 013777 001102 164576 MOV @#0BUFSV,@RSBA ;SET UP CURRENT ADDRESS
2727 014314 012777 177700 164566 MOV #177700,@RSWC ;SET WORD COUNT
2728 014322 012703 172060 MOV #172060,R3 ;GET RSLA REG
2729 014326 011304 1$: MOV @R3,R4 ;WAIT FOR
2730 014330 042704 000017 BIC #17,R4 ;THE MIDDLE
2731 014334 022704 000020 CMP #20,R4 ;OF SECTOR 0
2732 014340 001372 BNE 1$ ;BEFORE DOING A SEARCH
2733 014342 012777 000031 164534 MOV #31,@RSCS1 ;SEARCH
2734 014350 012777 000000 164544 MOV #C,@RSAS ;TRY TO CAUSE ERROR
2735 014356 005037 001174 CLR WORK ;CLEAR COUNTER
2736 014362 032777 000200 164526 2$: BIT #BIT7,@RSDS ;WAIT FOR DRY
2737 014370 001004 BNE 3$ ;READY CONT
2738 014372 005237 001174 INC WORK ;COUNT
2739 014376 001371 BNE 2$ ;RETRY
2740 014400 104001 HLT !CS1 ;RDY NEVER CAME UP
2741 014402 005777 164512 3$: TST @RSER ;DID RMR SET?
2742 014406 001401 BEQ .+4 ;NO
2743 014410 104002 HLT !ER ;ER SHOULD = 0
2744 014412 022777 110600 164476 CMP #110600,@RSDS ;DID ERR SET?
2745 014420 001401 BEQ .+4 ;NO
2746 014422 104042 HLT !DS!ER ;DS SHOULD = 110600 -
2747 014424 022777 104230 164452 CMP #104230,@RSCS1 ;IS CS1 CORRECT?
2748 014432 001401 BEQ .+4 ;YES
2749 014434 104040 HLT !DS ;CS1 SHOULD = 144260
2750 014436 104414 CLRDK ;CLEAR ALL RS REG
2751 014440 022777 004200 164436 CMP #4200,@RSCS1 ;IS CS1 CORRECT?
2752 014446 001401 BEQ .+4 ;YES
2753 014450 104040 HLT !DS ;NO

```

```

2754 :*****
2755 :TEST 77 TEST DCK IN RSER
2756 :*****
2757 014452 104400 TST77: SCOPE
2758
2759 ;DO A WRITE AND THEN A CLEAR FUNCTION THAT SHOULD CAUSE DCK TO SET
2760
2761 014454 104414 DCKTST: CLRDK ;CLEAR ALL RS REG
2762 014456 022737 000004 0G1162 CMP #4,RS04DT ;IS THIS A LA DISK?
2763 014464 001004 BNE 7$ ;NO
2764 014466 012737 177640 001200 MOV #177640,WORK2 ;GET WC FOR LA DISK
2765 014474 000411 BR 1$ ;CONTINUE
2766 014476 012737 177500 001200 7$: MOV #177500,WORK2 ;GET WC FOR RS04
2767 014504 005737 001162 TST RS04DT ;IS THIS A RS04?
2768 014510 001003 BNE 1$ ;YES
2769 014512 012737 177600 001200 MOV #177600,WORK2 ;NO
2770 014520 013777 001200 164362 1$: MOV WORK2,@RSWC ;LOAD WC
2771 014526 012737 177777 027454 MOV #-1,OUTBUF ;WRITE -1
2772 014534 013777 001102 164350 MOV @#OBUFSV,@RSBA ;SET UP CURRENT ADDRESS
2773 014542 052777 000010 164336 4$: BIS #10,@RSCS2 ;SET BAI BIT
2774 014550 012777 000061 164326 MOV #61,@RSCS1 ;GO WRITE
2775 014556 105777 164322 5$: TSTB @RSCS1 ;IS RDY SET?
2776 014562 100375 BPL 5$ ;WAIT FOR WRITE TO FINISH
2777 014564 005077 164324 2$: CLR @RSDA ;SET DSK ADDRESS TO 0
2778 014570 005037 027454 CLR OUTBUF ;WRITE 0
2779 014574 013777 001102 164310 MOV @#OBUFSV,@RSBA ;SET UP CURRENT ADDRESS
2780 014602 012777 177777 164300 MOV #-1,@RSWC ;LOAD WC
2781 014610 012702 172060 MOV #172060,R2 ;PUT RSLA ADDR INTO R2
2782 014614 011203 3$: MOV (R2),R3 ;GET LA AND WAIT FOR
2783 014616 042703 000077 BIC #77,R3 ;SECTOR 40
2784 014622 022703 004000 CMP #4000,R3 ;BEFORE
2785 014626 001372 BNE 3$ ;WRITING
2786 014630 012777 000061 164246 MOV #61,@RSCS1 ;GO WRITE
2787 014636 011203 6$: MOV (R2),R3 ;GET RSLA AND WAIT FOR
2788 014640 042703 000017 BIC #17,R3 ;MIDDLE OF SECTOR
2789 014644 022703 000020 CMP #20,R3 ;0 BEFORE EXECUTING
2790 014650 001372 BNE 6$ ;A CLEAR FUNCTION
2791 014652 012777 000040 164226 MOV #40,@RSCS2 ;CLEAR ALL REG. DO IT THIS WAY
2792 014660 013777 001152 164220 MOV UNNUM,@RSCS2 ;DO NOT USE TRAP
    
```

2793	014666	105777	164212		INCW:	TSTB	@RSCS1	:IS BUSY CLEARED
2794	014672	100401				BMI	6\$	:FLAG CLEARED
2795	014674	104001				HLT	!CS1	:RDY NEVER CAME UP
2796	014676	013777	001102	164206	6\$:	MOV	@#OBUFSV,@RSBA	:SET UP CURRENT ADDRESS
2797	014704	013777	001200	164176		MOV	WORK2,@RSWC	:LOAD WC
2798	014712	012777	000071	164164		MOV	#71,@RSCS1	:GO READ
2799	014720	105777	164160		5\$:	TSTB	@RSCS1	:IS RDY SET?
2800	014724	100375				BPL	5\$	:WAIT FOR READ TO FINISH
2801	014726	022777	100000	164164		CMP	#100000,@RSER	:DID DCK SET?
2802	014734	001401				BEQ	+.4	:YES
2803	014736	104002				HLT	!ER	:DCK DID NOT SET
2804	014740	022777	150600	164150		CMP	#150600,@RSDS	:DID ERR SET?
2805	014746	001401				BEQ	+.4	:YES
2806	014750	104040				HLT	!DS	:ER DID NOT SET BY DCK
2807	014752	022777	144270	164124		CMP	#144270,@RSCS1	:IS CS1 CORRECT?
2808	014760	001401				BEQ	+.4	:YES
2809	014762	104044				HLT	!DS!DA	
2810	014764	017700	164116			MOV	@RSCS2,BAD	:GET CS2
2811	014770	013701	001152			MOV	UNNUM,GOOD	:GET UNIT #
2812	014774	052701	000100			BIS	#100,GOOD	:SET IR
2813	015000	020100				CMP	GOOD,BAD	:IS CS2 CORRECT?
2814	015002	001401				BEQ	+.4	:YES
2815	015004	104000				HLT		
2816	015006	012701	177700			MOV	#177700,GOOD	:NO
2817	015012	017700	164072		1\$:	MOV	@RSWC,BAD	:DID TRANSFER STOP AT END OF SECTOR?
2818	015016	020100				CMP	GOOD,BAD	
2819	015020	001401				BEQ	+.4	:YES
2820	015022	104000				HLT		:NO
2821	015024	012701	027454			MOV	#OUTBUF,GOOD	:GET BA
2822	015030	022737	000004	001162		CMP	#4,RS04DT	:LA DISK?
2823	015036	001003				BNE	7\$	:NO
2824	015040	062701	000100			ADD	#100,GOOD	:YES
2825	015044	000410				BR	3\$	:CONTINUE
2826	015046	005737	001162		7\$:	TST	RS04DT	:RS04?
2827	015052	001003				BNE	2\$	:YES
2828	015054	062701	000200			ADD	#200,GOOD	:RS03
2829	015060	000402				BR	3\$	
2830	015062	062701	000400		2\$:	ADD	#400,GOOD	:GET CORRECT ANS FOR BA
2831	015066	017700	164020		3\$:	MOV	@RSBA,BAD	:GET BA
2832	015072	020001				CMP	BAD,GOOD	:IS BA CORRECT?
2833	015074	001401				BEQ	+.4	:YES
2834	015076	104000				HLT		:NO
2835	015100	104414				CLRDK		:CLEAR ALL RS REG
2836	015102	005777	164012			TST	@RSER	:DID DCK CLEAR?
2837	015106	001401				BEQ	+.4	:YES
2838	015110	104002				HLT	!ER	:DCK DID NOT CLEAR WITH CLEAR
2839	015112	012777	177500	163770		MOV	#177500,@RSWC	:CLEAR DCK ON
2840	015120	013777	001102	163764		MOV	@#OBUFSV,@RSBA	:DRIVE BY WRITING
2841	015126	012777	000061	163750		MOV	#61,@RSCS1	:GOOD DATA
2842	015134	105777	163744		4\$:	TSTB	@RSCS1	:ON DRIVE
2843	015140	100375				BPL	4\$	

```

2844 ;TEST THE ABILITY OF THE DISK CONTROL TO
2845 ;INCREMENT THE TRACK REGISTER.
2846
2847 ;A ONE WORD WRITE WILL BE EXECUTED
2848 ;RSDA=7777 RSWC = -1
2849 ;AT THE COMPLETION OF THE WRITE RSDA = 10000
2850 ;*****
2851 ;TEST 100 TEST DISK ADDRESS REGISTER
2852 ;*****
2853 015142 104400 TST100: SCOPE
2854
2855 015144 104414 DKADR: CLRDK ;CLEAR ALL RS REG
2856 015146 012777 177777 163734 MOV #177777,@RSWC ;SET WORD COUNT TO -1
2857 015154 013777 001102 163730 MOV @#OBUFSV,@RSBA ;SET UP CURRENT ADDRESS
2858 015162 012777 007777 163724 MOV #7777,@RSDA ;SET RSDA TO ALL ONES
2859 015170 012777 000061 163706 MOV #61,@RSCS1 ;GO WRITE ONE WORD
2860 015176 004737 026670 JSR PC, WAITRY ;WAIT FOR READY
2861 015202 104001 HLT !CS1 ;RDY DID NOT COME UP
2862 015204 027727 163704 010000 5$: CMP @RSDA,#10000 ;DOES RSDA=0
2863 015212 001401 BEQ .+4 ;RSDA OK
2864 015214 104004 HLT !DA ;DA DID NOT INCREMENT
2865
2866 ;*****
2867 ;TEST 101 TEST IAE ERROR
2868 ;*****
2869 015216 104400 TST101: SCOPE
2870
2871 ;IAE ERROR SHOULD SET ERR,ATA AND SC BITS
2872
2873 015220 104414 IAERR: CLRDK ;CLEAR ALL RS REG
2874 015222 012777 177777 163660 MOV #177777,@RSWC ;SET WC TO -1
2875 015230 013777 001102 163654 MOV @#OBUFSV,@RSBA ;SET UP BUS ADDRESS
2876 015236 012777 017777 163650 MOV #17777,@RSDA ;SET DA TO RECEIVE ERROR
2877 015244 012777 000061 163632 MOV #61,@RSCS1 ;GO WRITE ONE WD
2878 015252 105777 163626 7$: TSTB @RSCS1 ;TEST FOR ERR OR RDY
2879 015256 100401 BMI .+4 ;OK CONT.
2880 015260 000774 BR 7$ ;WAIT
2881 015262 022777 002000 163630 CMP #2000,@RSER ;DID IAE SET?
2882 015270 001401 BEQ .+4 ;YES
2883 015272 104002 HLT !ER ;IAE SHOULD BE SET
2884 015274 022777 150600 163614 CMP #150600,@RSDS ;DID ERR SET?
2885 015302 001401 BEQ .+4 ;YES
2886 015304 104140 HLT !DS!AS ;ERR SHOULD BE SET
2887 015306 022777 144260 163570 CMP #144260,@RSCS1 ;DID SC SET?
2888 015314 001401 BEQ .+4 ;YES
2889 015316 104001 HLT !CS1 ;SC SHOULD BE SET
2890 015320 104414 CLRDK ;CLEAR ALL RS REG
2891 015322 005777 163572 TST @RSER ;CLR ERRORS?
2892 015326 001401 BEQ .+4 ;YES
2893 015330 104002 HLT !ER ;ERR DID NOT CLR WITH 40 IN CS2
    
```



```

2894 ;IN THIS ROUTINE THE PROGRAM WILL GENERATE A
2895 ;NON-EXISTENT DISK ERROR
2896
2897 :*****
2898 ;TEST 102 TEST FOR NON-EXISTENT DISK ERROR
2899 :*****
2900 TST102: SCOPE
2901
2902 015332 104400 NEDTST: CLRDK ;CLEAR ALL RS REG
2903 015334 104414 MOV #401,WORK ;SET UP FOR N.E.D. NUMBER
2904 015336 012737 000401 001174 CLR GOOD ;LOOK FOR NON EXISTENT DRIVES
2905 015344 005001 001174 001154 1$: BIT WORK,UNITSV ;ON THE SYSTEM
2906 015346 033737 BEQ 3$ ;FOUND NON EXISTENT DRIVE
2907 015354 001405 INC GOOD ;CONTAINS UNIT #
2908 015356 005201 ROL WORK ;KEEP LOOKING
2909 015360 006137 001174 BCS NEDDON ;COULD NOT FIND ANY NON EXISTENT DRIVES
2910 015364 103452 BR 1$ ;LOOK FOR NED
2911 015370 010177 163512 3$: MOV GOOD,@RSCS2 ;LOAD NED IN CS2
2912 015374 005077 163514 CLR @RSDA ;WRITE DRIVE REG
2913 015400 005777 163514 TST @RSER ;DID ANY BITS SET IN RSER?
2914 015404 001401 BEQ .+4 ;NO
2915 015406 104040 HLT !DS ;WHY DID RSER CHANGE?
2916 015410 017700 163472 MOV @RSCS2,BAD ;GET CS2
2917 015414 052701 010100 BIS #10100,GOOD ;SET NED AND IR
2918 015420 020100 CMP GOOD,BAD ;IS CS2 CORRECT?
2919 015422 001401 BEQ .+4 ;YES
2920 015424 104000 HLT ;GOOD=CORRECT CS2 BAD=CS2
2921 015426 022777 160200 163450 CMP #160200,@RSCS1 ;IS CS1 CORRECT?
2922 015434 001401 BEQ .+4 ;YES
2923 015436 104200 HLT !CS2 ;TRE SHOULD SET BY NED ERROR
2924 015440 005777 163456 TST @RSAS ;DID ANY BITS SET?
2925 015444 001401 BEQ .+4 ;NO
2926 015446 104100 HLT !AS ;WHY DID AT BITS SET?
2927 015450 112777 000100 163462 MOVB #100,@RSCS1B ;CLEAR TRE
2928 015456 032777 010000 163422 BIT #NED,@RSCS2 ;DID NED CLEAR
2929 015464 001401 BEQ .+4 ;YES
2930 015466 104200 HLT !CS2 ;NED DID NOT CLEAR
2931 015470 017737 163420 001174 MOV @RSDA,WORK ;READ DRIVE REG
2932 015476 032777 010000 163402 BIT #NED,@RSCS2 ;DID NED SET?
2933 015504 001001 BNE .+4
2934 015506 104040 HLT !DS ;NED DID NOT SET
2935 015510 000431 BR NNDD ;GET OUT
2936 015512 032737 010000 001160 NEDDON: BIT #BIT12,ONCEE ;D WAS THIS TYPED BEFORE?
2937 015520 001025 BNE NNDD ;YES
2938 015522 104402 015526 TYPE .,+2 ;.ASCIZ <15><12>'COULD NOT FIND A NON-EXISTENT DRIVE'
2939 015574 052737 010000 001160 NNDD: BIS #BIT12,ONCEE ;SET TYPED FLAG
    
```

```

2940 :*****
2941 :TEST 103 TEST THAT DAO IN RSER AND LBT IN RSDS DO SET
2942 :*****
2943 015602 104400 TST103: SCOPE
2944
2945 015604 104414 DAOTST: CLRDK ;CLEAR ALL RS REG
2946 015606 022737 000004 001162 CMP #4,RS04DT ;RS03LA?
2947 015614 001004 BNE 3$ ;NO
2948 015616 012777 177737 163264 MOV #-41,@RSWC ;LOAD WORD COUNT
2949 015624 000411 BR 1$ ;CONT
2950 015626 012777 177577 163254 3$: MOV #-201,@RSWC ;LOAD WC FOR RS04
2951 015634 005737 001162 TST RS04DT ;IS THIS A RS04?
2952 015640 001003 BNE 1$ ;YES
2953 015642 012777 177677 163240 MOV #-101,@RSWC ;NO
2954 015650 012777 007777 163236 1$: MOV #7777,@RSDA ;SET RSDA=TO ALL ONES
2955 015656 013777 001102 163226 2$: MOV @#0BUF SV,@RSBA ;CURRENT ADDRESS=OUTBUF
2956 015664 012777 000061 163212 MOV #61,@RSCS1 ;WRITE
2957 015672 004737 026670 JSR PC,WAITRY ;WAIT FOR READY
2958 015676 104001 HLT !CS1 ;RDY DID NOT SET
2959 015700 022777 001000 163212 CMP #1000,@RSER ;DID DAO SET?
2960 015706 001401 BEQ .+4 ;YES
2961 015710 104002 HLT !ER ;DAO DID NOT SET
2962 015712 022777 152600 163176 CMP #152600,@RSDS ;DID LBT SET?
2963 015720 001401 BEQ .+4 ;YES
2964 015722 104040 HLT !DS ;LBT DID NOT SET
2965 015724 005777 163154 TST @RSCS1 ;IS ERROR FLAG SET
2966 015730 100401 BMI .+4 ;ERROR IS SET
2967 015732 104001 HLT !CS1 ;SC DID NOT SET
2968 015734 104414 CLRDK ;CLEAR ALL RS REG
2969 015736 022777 010600 163152 CMP #10600,@RSDS ;DID ATA +LBT CLEAR
2970 015744 001401 BEQ .+4 ;YES
2971 015746 104040 HLT !DS ;ATA DID NOT CLEAR BY CLR BIT
2972 015750 005777 163144 TST @RSER ;DID DAO CLEAR?
2973 015754 001401 BEQ .+4 ;YES
2974 015756 104002 HLT !ER ;DAO DID NOT CLEAR WITH CLEAR

```

```

2975 :*****
2976 :TEST 104 TEST THAT LBT DOES SET AND DAO DOES NOT
2977 :*****
2978 015760 104400 TST104: SCOPE
2979
2980 015762 104414 DAOTT: CLRDK ;CLEAR ALL RS REG
2981 015764 022737 000004 001162 CMP #4,RS04DT ;RS03LA?
2982 015772 001004 BNE 3$ ;NO
2983 015774 012777 177741 163106 MOV #-37,@RSWC ;LOAD WORD COUNT
2984 016002 000411 BR 1$ ;CONT
2985 016004 012777 177601 163076 3$: MOV #-177,@RSWC ;LOAD WC FOR RS04
2986 016012 005737 001162 TST RS04DT ;IS THIS A RS04?
2987 016016 001003 BNE 1$ ;YES
2988 016020 012777 177701 163062 MOV #-77,@RSWC ;NO
2989 016026 012777 007777 163060 1$: MOV #7777,@RSDA ;SET RSDA=TO ALL ONES
2990 016034 013777 001102 163050 2$: MOV @#0BUF SV,@RSBA ;CURRENT ADDRESS=OUTBUF
2991 016042 012777 000061 163034 MOV #61,@RSCS1 ;WRITE
2992 016050 004737 026670 JSR PC,WAITRY ;WAIT FOR READY
2993 016054 104001 HLT !CS1 ;RDY DID NOT SET
2994 016056 005777 163036 TST @RSER ;ANY ERRORS?
2995 016062 001401 BEQ .+4 ;NO
2996 016064 104002 HLT !ER ;YES
2997 016066 022777 012600 163022 CMP #12600,@RSDS ;DID LBT SET?
2998 016074 001401 BEQ .+4 ;YES
2999 016076 104040 HLT !DS ;LBT DID NOT SET
3000 016100 005777 163000 TST @RSCS1 ;IS ERROR FLAG SET
3001 016104 100001 BPL .+4 ;NO
3002 016106 104001 HLT !CS1 ;ERROR
3003 016110 104414 CLRDK ;CLEAR ALL RS REG
3004 016112 022777 010600 162776 CMP #10600,@RSDS ;DID LBT CLEAR
3005 016120 001401 BEQ .+4 ;YES
3006 016122 104040 HLT !DS ;ATA DID NOT CLEAR BY CLR BIT
    
```

```

3007 :*****
3008 :TEST 105 EXECUTE FUNCTION WITH ERROR BITS SET
3009 :*****
3010 016124 104400 TST105: SCOPE
3011
3012 016126 104414 ERTST: CLRDK ;CLEAR ALL RS REG
3013 016130 012777 177017 162762 MOV #177017,@RSER ;LOAD ER
3014 016136 017700 162760 MOV @RSAS,BAD ;GET AS REG
3015 016142 013701 001156 MOV UNCMP,GOOD ;GET UNIT ATA BIT
3016 016146 042701 177400 BIC #177400,GOOD ;CLEAR JUNK
3017 016152 020100 CMP GOOD,BAD ;IS AS REG CORRECT?
3018 016154 001401 BEQ .+4 ;YES
3019 016156 104100 HLT !AS ;AS BIT SHOULD BE SET
3020 016160 022777 104200 162716 CMP #104200,@RSCS1 ;DID ERRS SET IN CS1?
3021 016166 001401 BEQ .+4 ;YES
3022 016170 104040 HLT !DS ;CS1 SHOULD =104200
3023 016172 013777 001156 162722 MOV UNCMP,@RSAS ;CLEAR ATA BIT
3024 016200 005777 162716 TST @RSAS ;DID IT CLEAR?
3025 016204 001401 BEQ .+4 ;YES
3026 016206 104100 HLT !AS ;COULD NOT CLEAR AS BIT
3027 ;BY LOADING A 1 INTO IT
3028 016210 022777 004200 162666 CMP #4200,@RSCS1 ;DID SC CLEAR BY
3029 016216 001401 BEQ .+4 ;CLEARING ATA
3030 016220 104002 HLT !ER ;NO
3031 016222 012737 177777 027454 MOV #177777,OUTBUF ;DATA TO BE XFERED
3032 016230 013777 001102 162654 MOV @#OBUFSV,@RSBA ;SET UP CURRENT ADDRESS
3033 016236 012777 177777 162644 MOV #-1,@RSWC ;LOAD WC WITH -1
3034 016244 012777 000071 162632 MOV #71,@RSCS1 ;DO READ FUNCTION
3035 016252 032777 000001 162624 BIT #1,@RSCS1 ;DID GO BIT CLEAR
3036 016260 001401 BEQ .+4 ;YES
3037 016262 104001 HLT !CS1 ;GO BIT SHOULD BE CLEARED
3038 016264 105777 162614 1$: TSTB @RSCS1 ;WAIT FOR READY
3039 016270 100375 BPL 1$ ;WAIT
3040 016272 022777 144270 162604 CMP #144270,@RSCS1 ;DID ERRS CLEAR BY SETTING GO BIT?
3041 016300 001401 BEQ .+4 ;YES
3042 016302 104002 HLT !ER ;NO
  
```

3043	016304	017700	162576		MOV	@RSCS2,BAD	:GET CS2
3044	016310	012701	001100		MOV	#1100,GOOD	:GET CORRECT ANS
3045	016314	053701	001152		BIS	UNNUM,GOOD	:GET UNIT #
3046	016320	020100			CMP	GOOD,BAD	:IS CS2 CORRECT?
3047	016322	001401			BEQ	+.4	:YES
3048	016324	104000			HLT		:GOOD = WHAT CS2 SHOULD =
3049	016326	022777	150600	162562	CMP	#150600,@RSDS	:DID ERR BITS SET?
3050	016334	001401			BEQ	+.4	:NO
3051	016336	104040			HLT	!DS	:ERR BIT SHOULD BE 1
3052	016340	022777	177777	162542	CMP	#-1,@RSWC	:DID WC MOVE?
3053	016346	001401			BEQ	+.4	:NO
3054	016350	104010			HLT	!WC	:WC SHOULD = 1777777
3055	016352	005777	162536		TST	@RSDA	:DID DA MOV
3056	016356	001401			BEQ	+.4	:NO
3057	016360	104004			HLT	!DA	:DA SHOULD =0
3058	016362	023777	001102	162522	CMP	@#0BUF SV,@RSBA	:DID BA MOVE
3059	016370	001401			BEQ	+.4	:NO
3060	016372	104020			HLT	!BA	:BA MOVED
3061	016374	033777	001156	162520	BIT	UNCMP,@RSAS	:AS SHOULD BE SET
3062	016402	001001			BNE	+.4	:IS IT?
3063	016404	104100			HLT	!AS	:NO
3064	016406	022777	177017	162504	CMP	#177017,@RSER	:DID ER CHANGE?
3065	016414	001401			BEQ	+.4	:NO
3066	016416	104002			HLT	!ER	:ER SHOULD NOT CHANGE

```

3067
3068
3069
3070 016420 104400
3071
3072 016422 104414
3073 016424 052777 000020 162454
3074 016432 005777 162472
3075 016436 017700 162444
3076 016442 012701 000120
3077 016446 053701 001152
3078 016452 020100
3079 016454 001401
3080 016456 104000
3081 016460 012777 000010 162442
3082 016466 022777 000010 162424
3083 016474 001401
3084 016476 104140
3085 016500 017700 162416
3086 016504 013701 001156
3087 016510 042701 177400
3088 016514 020100
3089 016516 001401
3090 016520 104100
3091 016522 022777 104200 162354
3092 016530 001401
3093 016532 104040
3094 016534 104414
3095 016536 022777 004200 162340
3096 016544 001401
3097 016546 104002
3098 016550 017700 162332
3099 016554 013701 001152
3100 016560 052701 000100
3101 016564 020100
3102 016566 001401
3103 016570 104000

:*****
:TEST 106 PAT AND MCPE TEST
:*****
TST106: SCOPE

PATST: CLRDK
BIS #BIT4,@RSCS2 ;CLEAR ALL RS REG
TST @RSMR ;SET PAT
MOV @RSCS2,BAD ;READ DRIVE REG
MOV #120,GOOD ;GET CS2
BIS UNNUM,GOOD ;MDPE SHOULD
CMP GOOD,BAD ;NOT SET
BEQ .+4 ;IS CS2 CORRECT?
HLT ;YES
MOV #10,@RSMR ;BAD = CS2 GOOD = CORRECT ANS
CMP #10,@RSER ;CAUSE PAR TO SET IN RSER
BEQ .+4 ;DID PAR SET?
HLT !AS!DS ;YES
MOV @RSAS,BAD ;GET AS REG
MOV UNCMP,GOOD ;GET UNIT ATA BIT
BIC #177400,GOOD ;CLEAR JUNK
CMP GOOD,BAD ;IS AS REG CORRECT?
BEQ .+4 ;YES
HLT !AS ;AS BIT SHOULD BE SET
CMP #104200,@RSCS1 ;DID ERRS SET IN CS1?
BEQ .+4 ;YES
HLT !DS ;CS1 SHOULD =104200
CLRDK ;CLEAR RS REG
CMP #4200,@RSCS1 ;IS CS1 CORRECT?
BEQ .+4 ;CLEARING ATA
HLT !ER ;NO
MOV @RSCS2,BAD ;CHECK TO SEE
MOV UNNUM,GOOD ;IF PAT CLEARS
BIS #100,GOOD
CMP GOOD,BAD
BEQ .+4
HLT ;PAT DID NOT CLEAR
  
```

```

3104 :*****
3105 :TEST 107          SET PAT BIT AND LOAD FUNCTION
3106 :*****
3107 016572 104400  TST107: SCOPE
3108
3109 016574 104414  SETPAT: CLRDK          :CLEAR ALL REG
3110 016576 052777 000010 162302  BIS      #BAI,@RSCS2        :SET BAI
3111 016604 012737 177777 027454  MOV      #177777,OUTBUF :DATA TO BE XFERED
3112 016612 013777 001102 162272  MOV      @#OBUFSV,@RSBA :SET UP CURRENT ADDRESS
3113 016620 012777 177000 162262  MOV      #-1000,@RSWC   :LOAD WC WITH -1
3114 016626 052777 000020 162252  BIS      #BIT4,@RSCS2   :SET PAT BIT
3115 016634 012777 000071 162242  MOV      #71,@RSCS1     :DO READ FUNCTION
3116 016642 105777 162236  1$:      TSTB      @RSCS1     :WAIT FOR READY
3117 016646 100375  BPL      1$           :WAIT
3118 016650 022777 144270 162226  CMP      #144270,@RSCS1 :DID CS1 GET LOADED?
3119 016656 001401  BEQ      .+4         :NO
3120 016660 104040  HLT      !DS         :IT SHOULD NOT
3121 016662 023777 001102 162222  CMP      @#OBUFSV,@RSBA :DID BA MOVE?
3122 016670 001401  BEQ      .+4         :NO
3123 016672 104020  HLT      !BA         :YES
3124 016674 022777 177000 162206  CMP      #-1000,@RSWC   :DID WC MOVE?
3125 016702 001401  BEQ      .+4         :NO
3126 016704 104010  HLT      !WC         :YES WHY?
3127
3128 :*****
3129 :TEST 110          DO FUNCTION THEN SET PAT BIT
3130 :*****
3131 016706 104400  TST110: SCOPE
3132 016710 104414  FUNDO: CLRDK          :CLEAR ALL REG
3133 016712 052777 000010 162166  BIS      #BAI,@RSCS2        :SET BAI
3134 016720 012737 177777 027454  MOV      #177777,OUTBUF :DATA TO BE XFERED
3135 016726 013777 001102 162156  MOV      @#OBUFSV,@RSBA :SET UP CURRENT ADDRESS
3136 016734 012777 177000 162146  MOV      #-1000,@RSWC   :LOAD WC WITH -1
3137 016742 012777 000071 162134  3$:      MOV      #71,@RSCS1     :DO A READ
3138 016750 100774  BMI      3$           :WAIT FOR BUSY
3139 016752 052777 000020 162126  BIS      #BIT4,@RSCS2   :SET PAT
3140 016760 105777 162120  2$:      TSTB      @RSCS1     :WAIT FOR READY
3141 016764 100375  BPL      2$           :
3142 016766 022777 144270 162110  CMP      #144270,@RSCS1 :DID MCPE SET?
3143 016774 001401  BEQ      .+4         :NO
3144 016776 104002  HLT      !ER         :YES
3145 017000 017700 162102  MOV      @RSCS2,BAD     :GET CS2
3146 017004 012701 000730  MOV      #730,GOOD     :GET CORRECT ANS
3147 017010 053701 001152  BIS      UNNUM,GOOD     :GET UNIT #
3148 017014 020100  CMP      GOOD,BAD      :IS CS2 CORRECT?
3149 017016 001401  BEQ      .+4         :YES
3150 017020 104000  HLT      :GOOD = WHAT CS2 SHOULD =
    
```

```

3151 017022 022777 010600 162066      CMP      #10600,@RSDS      ;DID ERR BITS SET?
3152 017030 001401                      BEQ      .+4              ;NO
3153 017032 104040                      HLT      !DS              ;ERR BIT SHOULD BE 1
3154 017034 005777 162060              TST      @RSER            ;IS ER CLEAR?
3155 017040 001401                      BEQ      .+4              ;YES
3156 017042 104044                      HLT      !DS!DA          ;NO ERRORS SHOULD BE SET
3157 017044 104414                      CLRDK                                ;CLEAR ALL RS REG
3158 017046 022777 004200 162030      CMP      #4200,@RSCS1    ;IS CS1 CORRECT?
3159 017054 001401                      BEQ      .+4              ;YES
3160 017056 104040                      HLT      !DS
3161                                     ;*****
3162                                     ;TEST 111                TEST PAR BY SETTING PAT
3163                                     ;*****
3164 017060 104400      TST111: SCOPE
3165
3166 017062 104414      PATTST: CLRDK            ;CLEAR ALL RS REG
3167 017064 052777 000010 162014      BIS      #BAI,@RSCS2     ;SET BAI
3168 017072 012737 177777 027454      MOV      #177777,OUTBUF  ;DATA TO BE XFERED
3169 017100 013777 001102 162004      MOV      @#OBUFSV,@RSBA ;SET UP CURRENT ADDRESS
3170 017106 012777 177000 161774      MOV      #-1000,@RSWC   ;LOAD WC WITH -1
3171 017114 012777 000061 161762      MOV      #61,@RSCS1     ;DO WRITE FUNCTION
3172 017122 105777 161756      1$: TSTB      @RSCS1     ;WAIT FOR READY
3173 017126 100775                      BMI      1$              ;WAIT
3174 017130 052777 000020 161750      BIS      #BIT4,@RSCS2   ;SET PAT
3175 017136 105777 161742      2$: TSTB      @RSCS1     ;WAIT FOR READY
3176 017142 100375                      BPL      2$
3177 017144 022777 144260 161732      CMP      #144260,@RSCS1 ;DID MCPE SET?
3178 017152 001401                      BEQ      .+4              ;NO
3179 017154 104002                      HLT      !ER              ;YES
3180 017156 017700 161724      MOV      @RSCS2,BAD      ;GET CS2
3181 017162 012701 000230      MOV      #230,GOOD       ;GET CORRECT ANS-- DO NOT CHECK IR - REASON 50 CYCLE
3182 017166 053701 001152      BIS      UNNUM,GOOD      ;GET UNIT #
3183 017172 042700 000100      BIC      #BIT6,BAD       ;CLEAR IR BIT FOR CS2 COMPARE
3184 017176 020100      CMP      GOOD,BAD        ;IS CS2 CORRECT?
3185 017200 001401                      BEQ      .+4              ;YES
3186 017202 104000                      HLT                                ;GOOD = WHAT CS2 SHOULD =
3187 017204 022777 150600 161704      CMP      #150600,@RSDS   ;DID ERR BITS SET?
3188 017212 001401                      BEQ      .+4              ;NO
3189 017214 104040                      HLT      !DS              ;ERR BIT SHOULD BE 1
3190 017216 022777 000010 161674      CMP      #10,@RSER       ;DID PAR SET?
3191 017224 001401                      BEQ      .+4              ;YES
3192 017226 104044                      HLT      !DS!DA          ;NO
3193 017230 104414                      CLRDK                                ;CLEAR ALL RS REG
3194 017232 022777 004200 161644      CMP      #4200,@RSCS1    ;IS CS1 CORRECT?
3195 017240 001401                      BEQ      .+4              ;YES
3196 017242 104040                      HLT      !DS
3197 017244 005777 161650      TST      @RSER            ;DID PAR CLEAR?
3198 017250 001401                      BEQ      .+4              ;YES
3199 017252 104040                      HLT      !DS
    
```



```

3200 ;*****
3201 ;TEST 112 TEST THE ABILITY TO FILL THE LAST SECTOR
3202 ;*****
3203 017254 104400 TST112: SCOPE
3204
3205 017256 104414 LASTSC: CLRDK ;CLEAR ALL RS REG
3206 017260 012777 007777 161626 MOV #7777,@RSDA ;SET RSDA=TO ALL ONES
3207 017266 022737 000004 001162 CMP #4,RS04DT ;LA DISK?
3208 017274 001004 BNE 2$ ;NO
3209 017276 012777 177740 161604 MOV #-40,@RSWC ;LOAD WORD COUNT
3210 017304 000411 BR 1$ ;CONTINUE
3211 017306 012777 177700 161574 2$: MOV #-100,@RSWC ;WORD COUNT=-100
3212 017314 005737 001162 TST RS04DT ;IS THIS A RS04?
3213 017320 001403 BEQ 1$ ;NO
3214 017322 012777 177600 161560 MOV #-200,@RSWC ;YES
3215 017330 013777 001102 161554 1$: MOV @#OBUFSV,@RSBA ;CURRENT ADDRESS=OUTBUF
3216 017336 012777 000061 161540 MOV #61,@RSCS1 ;WRITE
3217 017344 004737 026670 JSR PC,WAITRY ;WAIT FOR READY
3218 017350 104001 HLT !CS1 ;RDY DID NOT SET
3219 017352 005777 161542 TST @RSER ;DID ANY ERROR BITS SET?
3220 017356 001401 BEQ .+4 ;NO
3221 017360 104002 HLT !ER ;GOT AN ERROR
3222 017362 022777 012600 161526 CMP #12600,@RSDS ;DID LBT SET?
3223 017370 001401 BEQ .+4 ;YES
3224 017372 104040 HLT !DS ;LBT DID NOT SET
3225 017374 005777 161504 TST @RSCS1 ;IS ERROR FLAG SET
3226 017400 100001 BPL .+4 ;ERROR IS SET
3227 017402 104001 HLT !CS1 ;SC DID NOT SET
3228 017404 104414 CLRDK ;CLEAR ALL RS REG
3229 017406 022777 010600 161502 CMP #10600,@RSDS ;DID ATA +LBT CLEAR
3230 017414 001401 BEQ .+4 ;YES
3231 017416 104040 HLT !DS ;ATA DID NOT CLEAR BY CLR BIT
    
```

```

3232
3233
3234
3235
3236
3237
3238
3239 017420 104400
3240 017422 104414
3241 017424 012737 177777 027454
3242 017432 013777 001102 161452
3243 017440 022737 000004 001162
3244 017446 001004
3245 017450 012777 177740 161432
3246 017456 000411
3247 017460 012777 177600 161422 4$:
3248 017466 005737 001162
3249 017472 001003
3250 017474 012777 177700 161406
3251 017502 052777 000010 161376 5$:
3252 017510 012777 000061 161366
3253 017516 105777 161362 3$:
3254 017522 100375
3255 017524 005077 161364
3256 017530 012737 177777 027454
3257 017536 013777 001102 161346
3258 017544 012777 177777 161336
3259 017552 052777 000010 161326
3260 017560 012777 000061 161316
3261 017566 105777 161312 1$:
3262 017572 100375
3263 017574 042777 000010 161304
3264 017602 005737 001162
3265 017606 001404
3266 017610 012737 000200 001174
3267 017616 000403
3268 017620 012737 000100 001174 7$:
3269 017626 013701 001102 8$:
3270 017632 012721 177777
3271 017636 005021 6$:
3272 017640 005337 001174
3273 017644 001574
3274 017646 005077 161242
3275 017652 013777 001102 161232
3276 017660 022737 000004 001162
3277 017666 001004
3278 017670 012777 177740 161212
3279 017676 000412
3280 017700 005737 001162 11$:
3281 017704 001404
3282 017706 012777 177600 161174
3283 017714 000403
3284 017716 012777 177700 161164 9$:
3285 017724 012777 000051 161152 10$:

```

```

:FILL SECTOR WITH ALL ONES.
:NOW WRITE 1ST WORD IN SECTOR
:TEST REMAINING 63 WORDS FOR 0

```

```

:*****
:TEST 113 TEST FOR ZERO'S IN PARTIAL FILLED SECTOR
:*****

```

```

TST113: SCOPE
SECT: CLRDK ;CLEAR ALL RS REG
MOV #-1,OUTBUF ;PUT -1INTO OUTBUF
MOV @#OBUFSV,@RSBA ;SET UP CURRENT ADDR
CMP #4,RS04DT ;RS03LA DISK?
BNE 4$ ;NO
MOV #-40,@RSWC ;LOAD WORD COUNT
BR 5$ ;CONTINUE
MOV #-200,@RSWC ;LOAD WC FOR RS04
TST RS04DT ;RS04?
BNE 5$ ;YES
MOV #-100,@RSWC ;SET WORD COUNT TO -100
BIS #BAI,@RSCS2 ;SET BAI BIT
MOV #61,@RSCS1 ;WRITE
TSTB @RSCS1 ;IS RDY SET?
BPL 3$ ;NO
CLR @RSDA ;SET DSK ADDRESS TO 0
MOV #-1,OUTBUF ;PUT 177777 INTO OUTBUF
MOV @#OBUFSV,@RSBA ;SET UP CURRENT ADDR
MOV #-1,@RSWC ;SET WORD COUNT TO -1
BIS #10,@RSCS2 ;SET BAI BIT
MOV #61,@RSCS1 ;WRITE
TSTB @RSCS1 ;IS RDY SET?
BPL 1$ ;NO
BIC #10,@RSCS2 ;CLEAR BAI BIT
TST RS04DT ;RS04?
BEQ 7$ ;NO
MOV #200,WORK ;YES
BR 8$ ;CONT
MOV #100,WORK ;SET UP BUFFER
MOV @#OBUFSV,R1 ;GET STARTING ADD OF BUF
MOV #-1,(R1)+ ;LOAD FIRST WD WITH -1
CLR (R1)+ ;LOAD REST WITH 0
DEC WORK ;DONE YET?
BNE 6$ ;NO
CLR @RSDA ;SET DSK ADDRESS TO 0
MOV @#OBUFSV,@RSBA ;SET UP CURRENT ADDR
CMP #4,RS04DT ;RS03LA DISK?
BNE 11$ ;NO
MOV #-40,@RSWC ;LOAD WORD COUNT
BR 10$ ;CONTINUE
TST RS04DT ;RS04?
BEQ 9$ ;NO
MOV #-200,@RSWC ;YES
BR 10$ ;CONT
MOV #-100,@RSWC ;SET WORD COUNT TO -100
MOV #51,@RSCS1 ;WRITE CHECK

```

```

3286 017732 032777 000200 161144 2$: BIT #200,@RSCS1 ;IS RDY SET?
3287 017740 001774 BEQ 2$ ;NO
3288 017742 013701 001152 MOV UNNUM,GOOD ;GET UNIT #
3289 017746 052701 000100 BIS #100,GOOD ;SET IR BIT
3290 017752 017700 161130 MOV @RSCS2,BAD ;GET CS2
3291 017756 020100 CMP GOOD,BAD ;IS CS2 CORRECT?
3292 017760 001401 BEQ +4 ;YES
3293 017762 104002 HLT !ER ;THERE WAS A WRITE CHECK ERROR
3294
3295
3296 :*****
3297 :TEST 114 IF MEMORY MANAGEMENT IS AVAILABLE CHECK THE EXTENDED MEMORY ADDR
3298 :*****
3298 017764 104400 TST114: SCOPE
3299
3300 EXTST: CLRDK ;CLEAR ALL RS REG.
3301 017766 104414 MOV TIMES,TIMSV ;SAVE LOOP #
3302 017770 013737 022504 001164 MOV #10,TIMES ;LOOP 10 TIMES
3303 017776 012737 000010 022504 MOV #EXTTRP,4 ;SETUP TIMEOUT TRAP
3304 020004 012737 020510 000004 MOV #340,6
3305 020012 012737 000340 000006 TST @#SR0 ;IF MEMORY MANAGEMENT IS NOT
3306 020020 005737 177572 ;AVAILABLE THE PROGRAM WILL TRAP
3307 ;AND TRANSFER TO END OF THE TEST
3308 020024 012737 020502 000004 MOV #EXTRP,4
3309 020032 012737 007600 172356 MOV #7600,@#KIPAR7 ;OPEN I/O REGISTERS
3310 020040 005037 172340 CLR @#KIPAR0 ;FREE FIRST 4K
3311 020044 012737 000200 172342 MOV #200,@#KIPAR1 ;ENABLE SECOND 4K
3312 020052 012737 002000 172344 MOV #2000,@#KIPAR2
3313 020060 012737 177406 172300 MOV #400*256.-400+UP+RW,@#KIPDR0 ;SET KIPDR0=RW UP 400 BLOCKS
3314 020066 012737 177406 172302 MOV #400*256.-400+UP+RW,@#KIPDR1 ;SET KIPDR1=RW UP 400 BLOCKS
3315 020074 012737 177406 172304 MOV #400*256.-400+UP+RW,@#KIPDR2 ;SET KIPDR2=RW UP 400 BLOCKS
3316 020102 012737 177406 172316 MOV #400*256.-400+UP+RW,@#KIPDR7 ;SET KIPDR7=RW UP 400 BLOCKS
3317 020110 012737 000001 177572 MOV #1,@#SR0 ;TURN ON MEMORY MANAGEMENT
3318 020116 012702 040000 MOV #40000,R2 ;R2 EQUALS BASE ADDR
    
```

3319	020122	012712	177777		7\$:	MOV	#177777,(R2)	:INSERT PATTERN INTO 200000
3320	020126	012777	177776	160754		MOV	#-2,@RSWC	:SETUP WORDCOUNT
3321	020134	012777	177777	160750		MOV	#177777,@RSBA	:SETUP BUS ADDR
3322	020142	012777	000061	160734		MOV	#61,@RSCS1	:WRITE TWO WORDS ON DISK. RSBA
3323								:STARTS AT 177777 TO FORCE CARRY
3324								:TO SET A16
3325	020150	105777	160730			TSTB	@RSCS1	:WAIT FOR READY
3326	020154	100375				BPL	.-4	
3327	020156	005777	160722			TST	@RSCS1	
3328	020162	100002				BPL	1\$	
3329	020164	104046				HLT	!ER!DA!DS	:STATUS ERROR AFTER 2 WORD WRITE
3330	020166	000447				BR	2\$	:USING MEX0
3331	020170	022777	004660	160706	1\$:	CMP	#4660,@RSCS1	:IS CS1 CORRECT
3332	020176	001402				BEQ	3\$	:YES
3333	020200	104002				HLT	!ER	:CS2 DID NOT COMPARE
3334	020202	000441				BR	2\$	
3335	020204	005012			3\$:	CLR	(R2)	:CLEAR LOCATION 200000
3336	020206	005077	160702			CLR	@RSDA	:SETUP DA
3337	020212	012777	177777	160672		MOV	#177777,@RSBA	:SETUP BA
3338	020220	012777	177776	160662		MOV	#-2,@RSWC	:SETUP WC
3339	020226	012777	000071	160650		MOV	#71,@RSCS1	:READ TWO WORDS INTO LOCATIONS
3340								:177777 AND 200000.
3341	020234	105777	160644			TSTB	@RSCS1	:WAIT FOR READY
3342	020240	100375				BPL	.-4	
3343	020242	005777	160636			TST	@RSCS1	:ANY ERRORS?
3344	020246	100002				BPL	4\$	:BRANCH IF NO
3345	020250	104002				HLT	!ER	:ERROR OFTER READING 2 WORDS
3346	020252	000415				BR	2\$	
3347	020254	022777	004670	160622	4\$:	CMP	#4670,@RSCS1	:IS CS1 CORRECT?
3348	020262	001402				BEQ	5\$	:YES
3349	020264	104002				HLT	!ER	:CS1 DID NOT COMPARE
3350	020266	000407				BR	2\$	:READ STARTING AT 177777
3351	020270	022712	177777		5\$:	CMP	#177777,(R2)	:WAS DATA READ INTO LOCATION
3352	020274	001404				BEQ	2\$	:200000 CORRECTLY? - BRANCH IF YES
3353	020276	012701	177777			MOV	#177777,GOOD	
3354	020302	011200				MOV	(R2),BAD	
3355	020304	104000				HLT		:DATA COMPARE ERROR AT 200000
3356	020306	000240			2\$:	NOP		

3357	020310	104414			EXTT1:	CLRDK			;CLEAR ALL REG
3358	020312	012737	004000	172344		MOV	#4000,@#KIPAR2		
3359	020320	012702	040000			MOV	#40000,R2		;R2 EQUALS THE BASE ADDR
3360	020324	012712	177777		7\$:	MOV	#177777,(R2)		;INSERT PATTERN INTO 400000
3361	020330	012777	177777	160554		MOV	#177777,@RSBA		;SETUP BUS ADDR
3362	020336	012777	177776	160544		MOV	#-2,@RSWC		;LOAD WC
3363	020344	012777	000461	160532		MOV	#461,@RSCS1		;SET BIT A16 AND WRITE
3364	020352	105777	160526			TSTB	@RSCS1		;WAIT FOR READY
3365	020356	100375				BPL	-4		
3366	020360	005777	160520			TST	@RSCS1		;ANY ERRORS?
3367	020364	100001				BPL	4\$		;BRANCH IF NO
3368	020366	104002				HLT	!ER		;ERROR AFTER READING 2 WORDS
3369	020370	022777	005260	160506	4\$:	CMP	#5260,@RSCS1		;IS CS1 CORRECT?
3370	020376	001401				BEQ	10\$		;BRANCH IF YES
3371	020400	104002				HLT	!ER		;NO CS1 DID NOT COMPARE
3372									;READ STARTING AT 377777
3373	020402	005012			10\$:	CLR	(R2)		;CLEAR LOCATION 400000
3374									;READ TWO WORDS STARTING AT 377777
3375	020404	012777	177776	160476		MOV	#-2,@RSWC		;SETUP WC
3376	020412	005077	160476			CLR	@RSDA		;SETUP DA
3377	020416	012777	177777	160466		MOV	#177777,@RSBA		
3378	020424	012777	000471	160452		MOV	#471,@RSCS1		;CLEAR A 17 SET A16,READ
3379	020432	105777	160446			TSTB	@RSCS1		;WAIT FOR READY
3380	020436	100375				BPL	-4		
3381	020440	005777	160440			TST	@RSCS1		;ANY ERRORS?
3382	020444	100002				BPL	11\$		;BRANCH IF NO
3383	020446	104002				HLT	!ER		;ERROR WHILE READING TWO WORDS
3384	020450	000414				BR	EXTRP		
3385	020452	022777	005270	160424	11\$:	CMP	#5270,@RSCS1		;IS CS1 CORRECT?
3386	020460	001401				BEQ	12\$		;BRANCH IF YES
3387	020462	104002				HLT	!ER		;CS1 DID NOT COMPARE
3388									;READ STARTING AT 377777
3389	020464	022712	177777		12\$:	CMP	#177777,(R2)		;WAS DATA READ INTO LOCATION 400000
3390	020470	001404				BEQ	EXTRP		;CORRECTLY? - BRANCH IF YES
3391	020472	012701	177777			MOV	#177777,GOOD		
3392	020476	011200				MOV	(R2),BAD		
3393	020500	104000				HLT			;DATA COMPARE ERROR AT 400000 IF
3394									;RECEIVED=0 - LOCATION WASN'T ACCESSED
3395	020502	005037	177572		EXTRP:	CLR	@#SRO		;TURN OFF MEMORY MANAGEMENT
3396	020506	000401				BR	EXT1		
3397	020510	000240			EXTTRP:	NOP			;UPDATE TEST NUMBERS
3398	020512	012706	000500		EXT1:	MOV	#500,SP		;RESTORE STACK
3399	020516	012737	000006	000004	MEMOUT:	MOV	#6,4		
3400	020524	005037	000006			CLR	6		

```

3401 :*****
3402 :TEST 115 TEST PROGRAM INTERRUPT BY MOVING 300 INTO RSCS1
3403 :*****
3404 020530 104400 TST115: SCOPE
3405
3406 020532 104414 QES: CLRDK ;CLEAR ALL DRIVES
3407 020534 012706 000500 MOV #500,SP ;SETUP STACK
3408 020540 012777 020612 160366 MOV #PGTRAP,@RSVEC ;SET UP VECTOR
3409 020546 012777 000340 160362 MOV #340,@RSVCPS ;SET TRAP PS
3410 020554 012737 000200 177776 MOV #200,@#PS ;SET PS AT PRIORITY 4
3411 020562 012777 000300 160314 MOV #300,@RSCS1 ;THIS SHOULD CAUSE A TRAP
3412 020570 012737 000500 001174 MOV #500,WORK ;SETUP LOOP
3413 020576 005337 001174 1$: DEC WORK ;DEC LOOP SHOULD
3414 020602 001375 BNE 1$ ;INTERRUPE BEFORE LOOP IS DONE
3415 020604 104001 HLT !CS1 ;SHOULD NEVER GET HERE
3416 020606 000137 020626 JMP QESDON ;GET OUT
3417
3418 020612 022626 PGTRAP: CMP (6)+,(6)+ ;TRAP OK
3419 020614 022777 004200 160262 CMP #4200,@RSCS1 ;DID IE CLEAR?
3420 020622 001401 BEQ +4 ;YES
3421 020624 104001 HLT !CS1 ;IE SHOULD BE CLEARED
3422 020626 QESDON:
3423
3424 :*****
3425 :TEST 116 TEST THAT DISK DOES NOT INTERRUPT WHEN PS IS AT 5
3426 :*****
3427 020626 104400 TST116: SCOPE
3428
3429 020630 012706 000500 INTR5: MOV #500,SP ;SETUP STACK
3430 020634 013737 001164 022504 MOV TIMSV,TIMES ;RESTORE LOOP COUNTER
3431 020642 104414 CLRDK ;CLEAR ALL RS REG
3432 020644 012777 020724 160262 MOV #INT112,@RSVEC ;SET UP INTERRUPT VECTOR
3433 020652 012777 000340 160256 MOV #340,@RSVCPS ;SET PRIO.
3434 020660 012737 000240 177776 MOV #240,@#PS ;LOCK OUT ALL INTERRUPTS ABOVE
3435 020666 013700 177776 MOV @#PS,BAD ;GET PS
3436 020672 012777 177777 160210 MOV #177777,@RSWC ;SET WORD COUNT TO -1
3437 020700 013777 001102 160204 MOV @#OBUFSV,@RSBA ;LOAD CURRENT ADDRESS
3438 020706 012777 000161 160170 MOV #161,@RSCS1 ;GO WRITE (INTERRUPT ENABLED)
3439 020714 004737 026670 JSR PC,WAITRY ;WAIT FOR READY
3440 020720 104001 HLT !CS1 ;NO RDY NEVER CAME UP
3441 020722 000403 BR INTDON ;RESTART ROUTINE
3442 ;PROCESSOR SHOULD NOT TRAP TO INT112
3443
3444 020724 012701 000240 INT112: MOV #240,GOOD ;WHAT PS SHOULD HAVE
3445 020730 104000 HLT ;GOOD = CORRECT ANS FOR PS
3446 020732 INTDON: ;DONE GET OUT
  
```

```

3447
3448
3449
3450 020732 104400
3451
3452 020734 012706 000500 INTR4: MOV #500,SP ;SETUP STACK
3453 020740 104414 CLRDK ;CLEAR ALL RS REG
3454 020742 012777 021036 160164 MOV #INT114,@RSVEC ;SET UP DISK TRAP VECTOR
3455 020750 012777 000340 160160 MOV #340,@RSVCPS ;SET PRIO.
3456 020756 012737 000200 177776 MOV #200,@#PS ;SET PROCESSOR TO PRIORITY 4
3457 020764 013700 177776 MOV @#PS,BAD ;GET PS
3458 020770 012701 000200 MOV #200,GOOD ;GET CORRECT PS
3459 020774 012777 177777 160106 MOV #177777,@RSWC ;SET WORD COUNT TO -1
3460 021002 013777 001102 160102 MOV @#OBUFSV,@RSBA ;LOAD CURRENT ADDRESS
3461 021010 012777 000161 160066 MOV #161,@RSCS1 ;WRITE (INTERRUPT ENABLE
3462 021016 005037 001174 CLR WORK
3463 021022 005237 001174 INC WORK ;WAIT FOR INTERRUPT TO OCCUR
3464 021026 001375 BNE -4
3465 021030 104000 HLT ;GOOD=CORRECT PS BAD=WRONG PS
3466 021032 104042 HLT
3467 021034 000405 BR DONINT ;CONT
3468 021036 022777 004260 160040 INT114: CMP #4260,@RSCS1 ;DID IE CLEAR?
3469 021044 001401 BEQ +4 ;YES
3470 021046 104001 HLT !CS1 ;WHY DID NOT IE CLEAR
3471 021050
3472
3473
3474
3475
3476 021050 104400
3477
3478 021052 012706 000500 ERINT: MOV #500,SP ;SETUP STACK
3479 021056 012737 000200 177776 MOV #200,@#PS ;SET PS AT PRI 4
3480 021064 012777 021142 160042 MOV #ERRINT,@RSVEC ;SET UP INTERRUPT ADD.
3481 021072 104414 CLRDK ;CLEAR ALL RS REG
3482 021074 012777 000340 160034 MOV #340,@RSVCPS ;SET PRIO.
3483 021102 012777 177777 160004 MOV #177777,@RSDA ;SET RSDA=TO ALL ONES
3484 021110 012777 177600 157772 MOV #177600,@RSWC ;WORD COUNT=-200
3485 021116 013777 001102 157766 MOV @#OBUFSV,@RSBA ;CURRENT ADDRESS=OUTBUF
3486 021124 012777 000161 157752 MOV #161,@RSCS1 ;WRITE
3487 021132 004737 026670 JSR PC,WAITRY ;WAIT FOR READY
3488 021136 104042 1$: HLT !ER!DS ;Y DIDN'T PGM INTERRUPT IS RDY SET?
3489 021140 000406 BR FINST ;GET OUT
3490 021142 022777 144260 157734 ERRINT: CMP #144260,@RSCS1 ;IS CS1 RIGHT?
3491 021150 001401 BEQ +4 ;YES
3492 021152 104042 HLT !ER!DS
3493 021154 022626 CMP (6)+,(6)+ ;CLEAR STACK
3494 021156
    
```

```

:*****
:TEST 120 TEST INTERRUPT ON ERROR
:*****
TST120: SCOPE
    
```

```

3495 :*****
3496 :TEST 121 DYNAMIC FUNCTION TEST
3497 :*****
3498 021156 104400 TST121: SCOPE
3499 :EXECUTE FUNCTION MODIFY UNIT # AND DO A DRIVE SEARCH
3500 :DRIVE SEARCH WILL ONLY BE DONE IF THERE ARE AT LEAST 2 DRIVES
3501 :2ND DRIVE MAY NOT BE TESTED YET SO IF THIS TEST FAILS CHECK 2ND DRIVE
3502 :BEFORE TRYING TO DEBUG THIS TEST
3503
3504 021160 104414 MODNUM: CLRDK ;CLEAR ALL RS REG
3505 021162 013737 022504 001164 MOV TIMES,TIMSV ;SAVE LOOP COUNT
3506 021170 012737 000010 022504 MOV #10,TIMES ;LOOP ONLY 10 TIMES
3507 021176 005037 001176 CLR WORK1 ;CLEAR WORK LOC.
3508 021202 005003 CLR R3
3509 021204 005004 CLR R4
3510 021206 012702 022134 MOV #DVTAB,R2 ;SETUP TABLE
3511 021212 012737 000401 001174 MOV #401,WORK ;SETUP TO TEST FOR MORE DRIVES
3512 021220 033737 001174 001154 7$: BIT WORK,UNITSV ;IS DRIVE ON SYSTEM?
3513 021226 001403 BEQ 6$ ;NO
3514 021230 020437 001152 CMP R4,UNNUM ;IS THIS THE SAME DRIVE?
3515 021234 001017 BNE 8$ ;NO
3516 021236 005204 6$: INC R4 ;UPDATE 2ND UNIT #
3517 021240 000241 CLC
3518 021242 006137 001174 ROL WORK ;CHECK FOR NEXT DRIVE
3519 021246 103364 BCC 7$ ;NOT DONE YET
3520 021250 032737 000010 001176 BIT #BIT3,WORK1 ;MULTI DRIVE?
3521 021256 001016 BNE 12$ ;YES
3522 021260 013705 001152 MOV UNNUM,R5 ;LOAD UNIT NO
3523 021264 005205 INC R5 ;CHANGE IT
3524 021266 042705 177770 BIC #177770,R5 ;CLEAR JUNK
3525 021272 000410 BR 12$
3526 021274 052737 000010 001176 8$: BIS #BIT3,WORK1 ;SET FOUND MULTI DRIVE
3527 021302 010422 MOV R4,(R2)+ ;LOAD UNIT # INTO TABLE
3528 021304 005203 INC R3 ;COUNT # OF DRIVES
3529 021306 010337 001172 MOV R3,SAVEE ;SAVE IT
3530 021312 000751 BR 6$
3531 021314 012777 021430 157612 12$: MOV #TSTVEC,@RSVEC ;SETUP INT. TRAP
3532 021322 012777 000340 157606 MOV #340,@RSVCPS ;SETUP PRIO.
3533 021330 005037 027454 CLR OUTBUF ;CLR TO READ INTO
3534 021334 013777 001102 157550 MOV @#OBUFSV,@RSBA ;SET UP CURRENT ADDRESS
3535 021342 012777 177000 157540 MOV #-1000,@RSWC ;SET WORD COUNT
3536 021350 012777 000060 157536 1$: MOV #60,@RSDA ;LOAD DA
3537 021356 012702 022134 MOV #DVTAB,R2 ;GET TABLE
3538 021362 012777 000161 157514 MOV #161,@RSCS1 ;GO WRITE
3539 021370 005703 TST R3 ;MORE THEN 1 DRIVE?
3540 021372 001003 BNE 13$ ;YES
3541 021374 010577 157506 MOV R5,@RSCS2 ;NO MODIFY UNIT #
3542 021400 000407 BR 14$
    
```



```

3543 021402 012277 157500      13$:  MOV      (R2)+,@RSCS2  ;LOAD UNIT#
3544 021406 012777 000131 157470  MOV      #131,@RSCS1  ;DO SEARCH
3545 021414 005303          DEC      R3           ;DONE ALL DRIVES YET?
3546 021416 001371          BNE     13$          ;NO
3547 021420 012737 000200 177776  14$:  MOV      #200,@#PS   ;ENABLE INTERRUPTS
3548 021426 000001          WTDV:  WAIT
3549 021430 013777 001152 157450  TSTVEC: MOV      UNNUM,@RSCS2 ;GET 1ST DRIVE
3550 021436 017700 157442  MOV      @RSCS1,BAD  ;GET CS1
3551 021442 042700 100000  MOV      #BIT15,BAD  ;CLEAR SC
3552 021446 012701 004260  MOV      #4260,GOOD  ;GET CORRECT ANS
3553 021452 020100          CMP     GOOD,BAD    ;IS CS1 CORRECT?
3554 021454 001402          BEQ     4$          ;NO! X-FER OK
3555 021456 104000          HLT
3556 021460 104140          HLT      !DS!AS
3557 021462 005777 157422      4$:  TST      @RSWC       ;TEST WC
3558 021466 001401          BEQ     .+4         ;WORD COUNT DID OVERFLOW
3559 021470 104010          HLT      !WC
3560 021472 013701 001152  MOV      UNNUM,GOOD  ;SHOULD = 0
3561 021476 052701 000100  BIS      #100,GOOD   ;GET CORRECT
3562 021502 017700 157400  MOV      @RSCS2,BAD  ;ANS OF CS2
3563 021506 020100          CMP     GOOD,BAD    ;GET CS2
3564 021510 001401          BEQ     .+4         ;IS CS2 CORRECT?
3565 021512 104000          HLT      !YES
3566 021514 017700 157372  MOV      @RSBA,BAD   ;GOOD = CORRECT ANS FOR CS2
3567 021520 013701 001102  MOV      @#OBUFSV,GOOD ;FETCH CURRENT ADDRESS
3568 021524 062701 002000  ADD      #2000,GOOD  ;WHAT RSBA SHOULD EQUAL
3569 021530 020001          CMP     BAD,GOOD    ;UPDATE IT
3570 021532 001401          BEQ     .+4         ;IS RSBA CORRECT
3571 021534 104000          HLT      !YES EXECUTE CONTINUE
3572 021536 022737 000004 001162  CMP      #4,RS04DT  ;RSBA FAILED TO INCREMENT
3573 021544 001006          BNE     2$          ;RS03LA?
3574 021546 022777 000100 157340  CMP      #100,@RSDA ;NO
3575 021554 001420          BEQ     3$          ;IS DA CORRECT?
3576 021556 104104          HLT      !DA!AS
3577 021560 000416          BR      3$          ;OK
3578 021562 005737 001162      2$:  TST      RS04DT     ;CONTINUE
3579 021566 001006          BNE     5$          ;IS THIS A RS04?
3580 021570 022777 000070 157316  CMP      #70,@RSDA  ;YES
3581 021576 001407          BEQ     3$          ;IS DA CORRECT?
3582 021600 104104          HLT      !DA!AS
3583 021602 000405          BR      3$          ;YES
3584 021604 022777 000064 157302  5$:  CMP      #64,@RSDA  ;DA NOT CORRECT
3585 021612 001401          BEQ     .+4         ;CONTINUE
3586 021614 104046          HLT      !DA!ER!DS ;WAS RSDA INCREMENTED
3587 021616 012777 040000 157260  3$:  MOV      #TRE,@RSCS1 ;RSDA SHOULD CONTAIN A 64
3588 021624 032737 000010 001176  BIT      #BIT3,WORK1 ;CLEAR ALL ERRORS IF ANY
3589 021632 001461          WTDV1  ;MULTI DRIVE?
3590 021634 012777 021660 157272  1$:  MOV      #TTVEC,@RSVEC ;NO
3591 021642 012716 021656          MOV     #WTDV2,(SP) ;SETUP INT FOR NEXT DRIVE
3592 021646 012777 000100 157230  MOV     #100,@RSCS1 ;GET WAIT
3593 021654 000002          RTI      ;SET IE
3594 021656 000001          WTDV2: WAIT ;RETURN
    
```

```

3595          ;SERVICE ROUTINE FOR SEARCH FUNCTIONS
3596 021660 005002          ITVEC: CLR      R2          ;CLEAR UNIT #
3597 021662 000241          CLC
3598 021664 012737 000401 001174      MOV      #401,WORK
3599 021672 033777 001174 157222 1$: BIT      WORK,@RSAS      ;DID THIS DRIVE INT?
3600 021700 001006          BNE      2$
3601 021702 005202          INC      R2          ;UPDATE UNIT #
3602 021704 000241          CLC
3603 021706 006137 001174          ROL      WORK
3604 021712 103367          BCC      1$
3605 021714 104100          HLT      !AS          ;WHY DID WE INT WITH NO ATA???
3606 021716 010277 157164          2$: MOV      R2,@RSCS2      ;GET DRIVE
3607 021722 022777 110600 157166      CMP      #110600,@RSDS  ;DID PIP CLEAR?
3608 021730 001401          BEQ      .+4          ;YES
3609 021732 104140          HLT      !DS!AS      ;PIP BIT DID NOT CLEAR
3610 021734 022777 104230 157142      CMP      #104230,@RSCS1 ;DID SC SET?
3611 021742 001401          BEQ      .+4          ;YES
3612 021744 104140          HLT      !AS!DS      ;SC DID NOT SET
3613 021746 005337 001172          DEC      SAVEE        ;COUNT # OF INT
3614 021752 001411          BEQ      WTDV1        ;DONE YET?
3615 021754 013777 001174 157140      MOV      WORK,@RSAS    ;CLEAR AS
3616 021762 012777 000100 157114      MOV      #100,@RSCS1   ;SET IE
3617 021770 012716 021656          MOV      #WTDV2,(SP)   ;RETURN TO WAIT
3618 021774 000002          RTI
3619 021776 012737 000340 177776      WTDV1: MOV      #340,@#PS
3620 022004 012706 000500          MOV      #500,SP      ;CLEAR STACK
3621 022010 013777 001136 157116      MOV      RSVCP5,@RSVEC ;RESTORE INT VECTOR
3622 022016 005077 157114          CLR      @RSVCPS
3623 022022 104400          MODDON: SCOPE        ;DONE
3624 022024 013737 001164 022504      MOV      TIMSV,TIMES  ;RESTORE LOOP COUNT
3625 022032 012737 000340 177776      MOV      #340,@#PS    ;RESTORE PS
3626 022040 012737 000001 001000      MOV      #1,ICNT      ;FUGE TEST NUMBERS
3627 022046 000137 001730          OUT:   JMP      @#TRYNX ;TEST NEXT DRIVE
3628          .SBTTL      $DONE - BELL AND SCOPE ROUTINE
3629
3630 022052 104400          DONE:   SCOPE        ;TERMINATING SCOPE FOR LOOPING
3631 022054 062737 000001 001006      ADD      #1,PCNT+2    ;ADD 1 TO THE PASS COUNT
3632 022062 005537 001004          ADC      PCNT        ;MAKE IT DOUBLE PREC.
3633 022066 032777 002000 156732      BIT      #SW10,@SWR   ;RING THE BELL?
3634 022074 001004          BNE      4$
3635 022076 104402 022102          TYPE      .,+2      ;.ASCIZ <BELL><177>
3636 022106 013700 000042          4$:   MOV      @#42,R0  ;GET MONITOR ADDRESS
3637 022112 001405          BEQ      $END1        ;IF NONE
3638 022114 000005          RESET
3639 022116 004710          $ENDAD: JSR      7,(0) ;GO TO MONITOR
3640 022120 000240 000240          240,240,240 ;SAVE ROOM FOR ACT11
3641 022126 000137 001202          $END1: JMP      @#BEGIN ;RETURN
3642
3643 022132 000000          .TBIT: 0            ;T BIT FLAG
3644
3645 022134 000010          DVTAB: .BLKW      10
    
```

```

3646          .SBTTL          $TYPE - TTY TYPEOUT ROUTINE
3647
3648          ;THIS ROUTINE IS USE TO TYPE ASCII MESSAGES ON THE TTY. THE
3649          ;CALL CAN BE IN ONE OF 3 FORMS: 1) 'TYPE ,ADR' - TYPES THE
3650          ;MESSAGE STARTING IN LOCATION 'ADR:' 2) 'TYPE ,CHAR' - TYPES
3651          ;THE ASCII 'CHAR', AND 3) 'PRINT <<15><12>'MESSAGE'> - TYPES
3652          ;THE MESSAGE WHICH IS INLINE ASCII. THE FILLER CHARACTER WHICH IS
3653          ;TYPED AFTER A LINE FEED IS IN FILCHR AND THE NUMBER OF FILLERS
3654          ;IS IN FILCHR+1.
3655
3656          022154 010446          .TYPE:  MOV      R4,-(6)          ;SAVE R4
3657          022156 010546          MOV      R5,-(6)          ;SAVE R5
3658          022160 017605 000004  MOV      @4(6),R5        ;GET ADDRESS TO BE TYPED
3659          022164 032705 177400  BIT      #177400,R5      ;IS IT A TYPED?
3660          022170 001002          BNE      1$              ;NO
3661          022172 016605 000004  MOV      4(6),R5        ;GET ADDRESS OF CHARACTER
3662          022176 105715          $:      TSTB     (R5)      ;TERMINATOR?
3663          022200 001423          BEQ      2$              ;GET OUT IF SO
3664          022202 122715 000012  CMPB    #12,(R5)        ;IS THE CHAR A LINE FEED
3665          022206 001012          BNE      4$              ;NO - GET OUT
3666          022210 113704 001015  MOVB    FILCHR+1,R4     ;GET THE FILL COUNT
3667          022214 113777 001014 156602 5$:  MOVB    FILCHR,@TPB     ;TYPE A FILLER
3668          022222 105777 156570  TSTB    @TPS            ;DONE YET?
3669          022226 100375          BPL      -4              ;NO - WAIT
3670          022230 005304          DEC      R4              ;DEC COUNT
3671          022232 001370          BNE      5$              ;LOOP UNTIL 0
3672          022234 112577 156564 4$:  MOVB    (R5)+,@TPB     ;LOAD AND TYPE THE CHARACTER
3673          022240 105777 156552  TSTB    @TPS            ;IS THE PRINTER READY
3674          022244 100375          BPL      -4              ;WAIT UNTIL IT IS
3675          022246 000753          BR       1$              ;GET THE NEXT CHARACTER
3676          022250 017646 000004 2$:  MOV      @4(6),-(6)     ;GET ADDRESS TO BE TYPED
3677          022254 062766 000002 000006  ADD      #2,6(6)        ;ADD 2 TO THE ADDRESS
3678          022262 022666 000004  CMP      (6)+,4(6)      ;IS IT .+2?
3679          022266 001006          BNE      3$              ;NO
3680          022270 062705 000002  ADD      #2,R5          ;ADD 2 TO THE ADDRESS
3681          022274 042705 000001  BIC      #1,R5          ;BACK UP TO AN EVEN BYTE
3682          022300 010566 000004  MOV      R5,4(6)        ;RESTORE ADDRESS
3683          022304 012605          3$:  MOV      (6)+,R5        ;RESTORE R5
3684          022306 012604          MOV      (6)+,R4        ;RESTORE R4
3685          022310 000002          RTI
  
```

```

3686          .SBTTL          $SCOPE - SCOPE LOOP HANDLER
3687
3688          ;THIS ROUTINE HANDLES THE ITERATIONS, LOOPING, ERROR
3689          ;LOOPING, AND THE DISPLAYING OF THE TEST NUMBER.
3690          ;"SCOPE" IS PLACED BETWEEN EACH SUBTEST IN THE TEST AND
3691          ;RECORDS THE STARTING ADDRESS OF THE SUBTEST IN 'LAD:'
3692
3693          022312 104416          .SCOPE: KBDIN          ;GO CHECK FOR ^G
3694          022314 032777 000400 156504          BIT          #SW8,@SWR          ;LOOP ON SPEC. TEST?
3695          022322 001404          BEQ          1$          ;NO LOOP ON SPEC. TEST
3696          022324 127737 156476 001000          CMPB          @SWR,ICNT          ;ON RIGHT TEST? *SW7-0*
3697          022332 0C1453          BEQ          .OVER          ;NOT RIGHT TEST
3698          022334 032777 040000 156464 1$:          BIT          #SW14,@SWR          ;LOOP ON TEST?
3699          022342 001045          BNE          .KIT          ;LOOP ON TEST IS SET
3700          022344 000416          BR          3$          ;SKIP - NOP FOR XOR TESTER
3701          022346 013746 000004          MOV          @#4,-(6)          ;PUSH @#4 ON STACK
3702          022352 012737 022372 000004          MOV          #4$,@#4          ;SET FOR TIMEOUT
3703          022360 005737 177060          TST          @#177060          ;ERROR ON XOR?
3704          022364 012637 000004          MOV          (6)+,@#4          ;POP STACK INTO @#4
3705          022370 000422          BR          .SVLAD          ;NO ERROR - GO TO NEXT TEST
3706          022372 022626          4$:          CMP          (6)+,(6)+          ;CLEAR STACK
3707          022374 012637 000004          MOV          (6)+,@#4          ;POP STACK INTO @#4
3708          022400 000426          BR          .KIT          ;ERROR - LOOP ON TEST
3709          022402 032777 004000 156416 3$:          BIT          #SW11,@SWR          ;KILL ITERATIONS
3710          022410 001012          BNE          .SVLAD          ;YES - KILL ITERATIONS
3711          022412 105737 001001          TSTB          ICNT+1          ;FIRST ONE?
3712          022416 001404          BEQ          2$          ;BRANCH IF FIRST
3713          022420 123737 022504 001001          CMPB          TIMES,ICNT+1          ;DONE?
3714          022426 003013          BGT          .KIT          ;BRANCH IF NOT
3715          022430 112737 000001 001001 2$:          MOVB          #1,ICNT+1          ;FIRST ITERATION
3716          022436 105237 001000          .SVLAD: INCB          ICNT          ;COUNT TEST NUMBERS
3717          022442 011637 001010          MOV          (6),LAD          ;SAVE LOOP ADDRESS
3718          022446 013777 001000 156354          MOV          ICNT,@DISPLAY          ;DISPLAY TEST NO. AND ITERATION COUNT
3719          022454 000002          RTI          ;RETURN
3720
3721          022456 105237 001001          .KIT:          INCB          ICNT+1          ;INC THE ITERATION COUNT
3722          022462 013777 001000 156340          .OVER:          MOV          ICNT,@DISPLAY          ;SET UP DISPLAY
3723          022470 005737 001010          TST          LAD          ;FIRST ONE?
3724          022474 001760          BEQ          .SVLAD          ;YES
3725          022476 013716 001010          MOV          LAD,(6)          ;FUDGE RETURN ADDRESS
3726          022502 000002          RTI          ;FIXES PS
3727
3728          022504 000100          TIMES:          100          ;RUN 100 TIMES
  
```

```

3729          .SBTTL          $HLT - HLT ROUTINE (ERROR TYPEOUT)
3730
3731          ;THIS ROUTINE PRINTS OUT ERROR MESSAGES STARTING WITH THE
3732          ;ADDRESS OF THE 'HLT'. IT ALSO COUNTS THE NUMBER OF ERRORS
3733          ;AND HAS THE CAPABILITY OF LOOPING ON ERROR, BELL ON ERROR,
3734          ;'HALT' ON ERROR, AND INHIBIT TYPEOUTS. AN OPTIONAL ARGUMENT
3735          ;(HLT+3) WILL BE PLACED IN '.HLTCT:' FOR ADITIONAL TYPEOUTS.
3736
3737 022506 104416          .HLT:  KBDIN          ;GO CHECK FOR ^G
3738 022510 032777 002000 156310  BIT          #SW10,@SWR          ;BELL ON ERROR?
3739 022516 001402          BEQ          1$          ;NO - SKIP
3740 022520 104402 000007          TYPE          ,BELL          ;RING BELL
3741 022524 005237 001002 1$:  INC          ERRORS          ;COUNT THE NUMBER OF ERRORS
3742 022530 032777 020000 156270  BIT          #SW13,@SWR          ;SKIP TYPEOUT IF SET
3743 022536 001025          BNE          2$          ;SKIP TYPEOUTS
3744 022540 104402 022544          TYPE          ..+2          ;.ASCIZ <15><12>
3745 022550 011637 001012          MOV          (6),HLTADR          ;PUT ADDRESS OF INSTRUCTION ON STACK
3746 022554 162737 000002 001012  SUB          #2,HLTADR          ;FUDGE ADDRESS
3747 022562 117737 156224 022644  MOVB         @HLTADR,.HLTCT          ;GET HLT ARGUMENT
3748 022570 013746 001012          MOV          HLTADR,-(6)          ;PUT HLTADR ON STACK
3749 022574 104404          TYPEO          ;TYPE STACK IN OCTAL
3750 022576 104402 022602          TYPE          ..+2          ;.ASCIZ ""
3751 022606 004737 026156          JSR          PC,RSREG          ;GO TO USER ERROR ROUTINE
3752 022612 005777 156210 2$:  TST          @SWR          ;HALT ON ERROR
3753 022616 100001          BPL          .+4          ;SKIP IF CONTINUE
3754 022620 000000          HALT          ;HALT ON ERROR!
3755 022622 032777 001000 156176  BIT          #SW9,@SWR          ;CHECK FOR INHIBIT LOOP ON ERROR
3756 022630 001003          BNE          3$          ;SKIP IF LOOP ON ERROR
3757 022632 105037 001001          CLRB         ICNT+1          ;CLEAR ITERATION COUNT
3758 022636 000002          RTI          ;RETURN
3759 022640 000137 022456 3$:  JMP          .KIT          ;LOOP ON TEST UNTIL NO ERRORS
3760
3761 022644 000000          .HLTCT: 0          ;HLT ARGUMENT
  
```

```

3762          .SBTTL          $OCTAL - OCTAL TYPEOUT ROUTINE
3763
3764          ;THIS ROUTINE IS USED TO TYPE AN OCTAL NUMBER ON THE TTY. IT WILL TYPE
3765          ;ALL 6 CHARACTERS, SUPPRESS LEADING ZEROES, OR TYPE THE
3766          ;16 BITS. IT IS CALLED VIA THE TYPOCT, TYPBIT, OR TYPOCS MACRO'S.
3767
3768 022646 012737 170101 023034 .TYPEB: MOV      #170101,.PR      ;SET BIT FLAG AND 16. CHARACTER COUNT
3769 022654 000411                BR          .PTIT          ;NOW TYPE IT IN BIT FORM
3770 022656 112737 000001 023034 .TYPEO: MOVB   #1,.PR          ;SET ZERO FILL SWITCH
3771 022664 000402                BR          .+6           ;SKIP
3772 022666 005037 023034          .TYPES: CLR      .PR          ;SUPPRESS LEADING ZERO'S
3773 022672 112737 177772 023035          MOVB     #-6,.PR+1      ;SET COUNT
3774 022700
3775 022700 010446                MOV      R4,-(6)        ;PUSH R4 ON STACK
3776 022702 010546                MOV      R5,-(6)        ;PUSH R5 ON STACK
3777 022704 016605 000010          MOV      10(6),R5      ;GET THE DATA
3778 022710 012704 023036          MOV      #.PR+2,R4     ;SET POINTER TO FIRST ASCII CHAR.
3779 022714 105014                CLRB     (4)           ;CLEAR FIRST BYTE
3780 022716 000411                BR          .PRF        ;ROTATE FIRST BIT
3781 022720 105014          .PRL:  CLRB     (4)           ;CLEAR BYTE OF CHARACTER
3782 022722 032737 000100 023034          BIT      #100,.PR     ;BIT TYPING MODE?
3783 022730 001004                BNE     .PRF          ;YES - SKIP 2 ROTATES
3784 022732 006105                ROL      R5           ;ROTATE BIT INTO C
3785 022734 106114                ROLB    (4)           ;PACK IT
3786 022736 006105                ROL      R5           ;ROTATE BIT INTO C
3787 022740 106114                ROLB    (4)           ;PACK IT
3788 022742 006105          .PRF:  ROL      R5           ;ROTATE BIT INTO C
3789 022744 106114                ROLB    (4)           ;PACK IT
3790 022746 105714                TSTB    (4)           ;IS IT ZERO?
3791 022750 001402                BEQ     .+6           ;SKIP INC
3792 022752 105237 023034          INCB     .PR          ;SET FILL SWITCH
3793 022756 105737 023034          TSTB    .PR          ;CHECK FILL SWITCH
3794 022762 001402                BEQ     .+6           ;SKIP BITSET
3795 022764 152724 000060          BISB    #'0,(4)+     ;MAKE INTO ASCII CHAR
3796 022770 105237 023035          INCB     .PR+1        ;INC COUNT
3797 022774 001351                BNE     .PRL          ;REPEAT
3798 022776 022704 023036          CMP     #.PR+2,R4     ;EMPTY BUFFER?
3799 023002 001002                BNE     .+6           ;SKIP IF NOT
3800 023004 112724 000060          MOVB    #'0,(4)+     ;LOAD 1 ZERO
3801 023010 105014                CLRB    (4)           ;NULL TERMINATOR
3802 023012 104402 023036          TYPE    .PR+2        ;TYPE IT
3803 023016 012605                MOV     (6)+,R5       ;POP STACK INTO R5
3804 023020 012604                MOV     (6)+,R4       ;POP STACK INTO R4
3805 023022 016666 000002 000004          MOV     2(6),4(6)     ;GET RID OF
3806 023030 012616                MOV     (6)+,(6)     ;DATA WORD
3807 023032 000002                RTI                    ;RETURN
3808
3809 023034 000012          .PR:  .BLKW  12      ;COUNT, SWITCH, AND OUTPUT BUFFER
    
```

```

3810          .SBTTL          $POWER - POWER DOWN AND UP ROUTINES
3811
3812          ;THIS IS THE POWER FAIL ROUTINE WHICH WILL SAVE ALL
3813          ;THE GENERAL REGISTERS AND USER DEFINED REGISTERS THEN
3814          ;WAIT FOR POWER TO GO DOWN AND BE RESTORED.
3815          ;IF THERE ISN'T ENOUGH TIME FOR SAVING ALL THE REGISTERS,
3816          ;THE PROGRAM WILL HALT AT '.ILLUP'.
3817
3818 023060 012777 023204 000124 .POWER: MOV      #.ILLUP,@.PUVEC ;SET FOR FAST UP
3819 023066 012777 000340 000120      MOV      #340,@.PUVEC$+2 ;PRIO:7
3820 023074 010046          MOV      R0,-(6) ;PUSH R0 ON STACK
3821 023076 010146          MOV      R1,-(6) ;PUSH R1 ON STACK
3822 023100 010246          MOV      R2,-(6) ;PUSH R2 ON STACK
3823 023102 010346          MOV      R3,-(6) ;PUSH R3 ON STACK
3824 023104 010446          MOV      R4,-(6) ;PUSH R4 ON STACK
3825 023106 010546          MOV      R5,-(6) ;PUSH R5 ON STACK
3826 023110 010637 023210      MOV      SP, .SAVR6 ;SAVE SP
3827 023114 012777 023124 000070      MOV      #.POWUP,@.PUVEC ;SET UP VECTOR
3828 023122 000000          HALT      ;WAIT FOR PF
3829
3830 023124 013706 023210      .POWUP: MOV      .SAVR6,SP ;GET SP
3831 023130 005001          CLR      R1 ;WAIT LOOP FOR THE TTY
3832 023132 005201          1$: INC     R1 ;WAIT FOR THE INC
3833 023134 001376          BNE     1$ ;OF WORD
3834 023136 012605          MOV      (6)+,R5 ;POP STACK INTO R5
3835 023140 012604          MOV      (6)+,R4 ;POP STACK INTO R4
3836 023142 012603          MOV      (6)+,R3 ;POP STACK INTO R3
3837 023144 012602          MOV      (6)+,R2 ;POP STACK INTO R2
3838 023146 012601          MOV      (6)+,R1 ;POP STACK INTO R1
3839 023150 012600          MOV      (6)+,R0 ;POP STACK INTO R0
3840 023152 012737 023060 000024      MOV      #.POWER,@#24 ;SET UP THE POWER DOWN VECTOR
3841 023160 012737 000340 000026      MOV      #340,@#26 ;PRIO:7
3842 023166 104402 023172          TYPE    ..+2 ;.ASCIZ <15><12>'POWER'
3843 023202 000002          RTI     ;RETURN
3844
3845 023204 000000          .ILLUP: HALT ;THE POWER UP SEQUENCE WAS STARTED
3846 023206 000776          BR      -2 ;BEFORE THE POWER DOWN WAS COMPLETE
3847
3848 023210 000000          .SAVR6: 0 ;PUT THE SP HERE
3849 023212 000024 000026          .PUVEC: 24,26 ;POWER UP VECTOR
    
```

```

3850          .SBTTL          $RDOCT - OCTAL INPUT ROUTINE
3851
3852          ;THIS ROUTINE CALLS RDLIN, INPUTS A LINE FROM THE TTY AND CONVERTS
3853          ;IT INTO AN OCTAL NUMBER WHICH IS THE FIRST WORD ON THE STACK.
3854
3855 023216 011646          .RDOCT: MOV      (6),-(6)          ;MOVE THE PC
3856 023220 016666 000004 000002 MOV      4(6),2(6)          ;MOVE THE PS
3857 023226 010146          MOV      R1,-(6)          ;PUSH R1 ON STACK
3858 023230 010246          MOV      R2,-(6)          ;PUSH R2 ON STACK
3859 023232 010346          MOV      R3,-(6)          ;PUSH R3 ON STACK
3860 023234 104412 4$: RDLIN          ;READ A LINE INTO INPUT
3861 023236 005001          CLR      R1          ;INIT DATA WORD
3862 023240 005037 027366          CLR      CTN          ;CLEAR COUNT WORD
3863 023244 012703 023472          MOV      #INPUT,R3          ;INIT POINTER
3864 023250 112302 1$: MOV#B      (3)+,R2          ;GET A BYTE
3865 023252 122702 000015          CMPB    #15,R2          ;WAS IT A CR?
3866 023256 001421          BEQ     2$          ;GET OUT IF YES
3867 023260 122702 000060          CMPB    #'0,R2          ;CHECK FOR 0 OR GREATER
3868 023264 003024          BGT     3$          ;ERROR - LESS THAN 0
3869 023266 122702 000067          CMPB    #'7,R2          ;CHECK FOR 7 OR LESS
3870 023272 002421          BLT     3$          ;ERROR - GREATER THAN 7
3871 023274 006002          ROR     R2          ;GET
3872 023276 006002          ROR     R2          ;INTO
3873 023300 006002          ROR     R2          ;POSITION
3874 023302 006101          ROL     R1          ;FIRST BIT
3875 023304 006102          ROL     R2          ;GET
3876 023306 006101          ROL     R1          ;SECOND BIT
3877 023310 006102          ROL     R2          ;GET
3878 023312 006101          ROL     R1          ;THIRD BIT
3879 023314 005237 027366          INC     CTN          ;YES HE TYPED SOMETHING
3880 023320 000753          BR      1$          ;LOOP
3881 023322 010166 000012 2$: MOV      R1,12(6)          ;SAVE THE RESULT
3882 023326 012603          MOV      (6)+,R3          ;POP STACK INTO R3
3883 023330 012602          MOV      (6)+,R2          ;POP STACK INTO R2
3884 023332 012601          MOV      (6)+,R1          ;POP STACK INTO R1
3885 023334 000002          RTI
3886
3887 023336          3$:
3888 023336 104402 023342          TYPE   ;,+2          ;.ASCIZ "'?'<15><12>
3889 023346 000732          BR      4$          ;TRY AGAIN
  
```



```

3890          .SBTTL          $RDLIN - TTY INPUT ROUTINE
3891
3892          ;THIS ROUTINE INPUTS A LINE TERMINATED BY A RETURN INTO ADDRESS
3893          ;INPUT AND RETURNS A LINE FEED. THE BUFFER HAS A NULL TERMINATOR
3894          ;INSTEAD OF THE RETURN. RUBOUTS ARE HANDLED BY RETYPING
3895          ;THE LINE. BUFFER OVERFLOW ERRORS LIKE A RUBOUT.
3896
3897          023350 010546          .RDLIN: MOV      R5,-(6)          ;SAVE R5
3898          023352 012705 023472 1$:      MOV      #INPUT,R5          ;GET ADDRESS
3899          023356 022705 023512 2$:      CMP      #INPUT+16.,R5      ;BUFFER FULL?
3900          023362 001423          BEQ      4$              ;YES - TYPE '?'
3901          023364 105737 177560          TSTB    @#177560          ;WAIT FOR
3902          023370 100375          BPL     .-4              ;A CHARACTER
3903          023372 113715 177562          MOVB    @#177562,(5)      ;GET CHARACTER
3904          023376 142715 000200          BICB    #200,(5)         ;GET RID OF JUNK
3905          023402 122715 000025          CMPB    #25,(5)         ;IS IT A ^U
3906          023406 001006          BNE     5$              ;BRANCH IF NOT
3907          023410 104402 023414          TYPE    ..+2            ;.ASCIZ '^U'<15><12>
3908          023422 000753          BR      1$              ;START OVER
3909          023424 122715 000177 5$:      CMPB    #177,(5)         ;IS IT A RUBOUT
3910          023430 001005          BNE     3$              ;SKIP IF NOT
3911          023432
3912          023432 104402 023436 4$:      TYPE    ..+2            ;.ASCIZ '?'<15><12>
3913          023442 000743          BR      1$              ;ZAP THE BUFFER AND LOOP
3914          023444 111527 000000 3$:      MOVB    (5),#0          ;SET UP FOR TYPING
3915          023450 104402 023446          TYPE    ,3$+2           ;ECHO IT
3916          023454 122725 000015          CMPB    #15,(5)+        ;CHECK FOR RETURN
3917          023460 001336          BNE     2$              ;LOOP IF NOT RETURN
3918          023462 104402 000012          TYPE    ,12            ;TYPE A LINE FEED
3919          023466 012605          MOV     (6)+,R5         ;RESTORE R5
3920          023470 000002          RTI                    ;RETURN
3921
3922          023472 000020          INPUT: .BLKB 16.        ;TTY INPUT AREA
3923          .SBTTL          $TRAP - TRAP HANDLER
3924
3925          ;THIS ROUTINE DECODES A TRAP CALL AND JUMPS TO THE APROPRATE
3926          ;SUBROUTINE. THE CALL IS A 'TRAP+N' WHERE N IS A MULTIPLE OF 2.
3927          ;THE 'SET' MACRO WILL CREATE THE TABLE NEEDED. IT HAS TO
3928          ;FOLLOW THIS MACRO.
3929
3930          023512 011646          .TRAP: MOV      (6),-(6)      ;GET ADDRESS OF TRAP +2
3931          023514 162716 000002          SUB     #2,(6)          ;MAKE IT ADDRESS OF TRAP
3932          023520 017616 000000          MOV     @ (6),(6)       ;GET TRAP INSTRUCTION
3933          023524 062716 117132          ADD     #.TRP+2-TRAP,(6) ;GET DATA AND MAKE IT AN OFFSET
3934          023530 013607          .TRP:  MOV     @ (6)+,PC     ;GO TO PROPER SUBROUTINE
3935
3936          023532 022312          .SCOPE          ;SCOPE = TRAP+0 (104400)
3937          023534 022154          .TYPE           ;TYPE = TRAP+2 (104402)
3938          023536 022656          .TYPE0          ;TYPE0 = TRAP+4 (104404)
3939          023540 022666          .TYPES          ;TYPES = TRAP+6 (104406)
3940          023542 023216          .RDOCT          ;RDOCT = TRAP+10 (104410)
3941          023544 023350          .RDLIN          ;RDLIN = TRAP+12 (104412)
3942          023546 026140          .CLRDK          ;CLRDK = TRAP+14 (104414)
3943          023550 027216          .KBDIN          ;KBDIN = TRAP+16 (104416)
3944          023552 027076          .SUSWR          ;SUSWR = TRAP+20 (104420)
3945          023554 027300          .CNTLU          ;CNTLU = TRAP+22 (104422)
    
```

```

3946 ;ROUTINE TO ALLOW THE OPERATOR TO SET BITS
3947 ;IN THE I/O REGISTERS VIA THE SWITCH REGISTER
3948
3949 ;WORD COUNT REGISTER
3950 023556 012706 000500 SRSWC: MOV #500,SP ;SET UP STACK FOR TRAP CALL
3951 023562 104416 1$: KBDIN ;CHECK THE WORLD
3952 023564 017777 155236 155316 MOV @SWR,@RSWC ;MOV SWR INTO WORD COUNT REG
3953 023572 017777 155312 155230 MOV @RSWC,@DISPLAY ;DISPLAY IN LIGHTS
3954 023600 000770 BR 1$
3955
3956 ;CURRENT ADDRESS REGISTER
3957 023602 012706 000500 SRSBA: MOV #500,SP ;INIT THE STACK
3958 023606 104416 1$: KBDIN ;CTW
3959 023610 017777 155212 155274 MOV @SWR,@RSBA ;MOV SWR INTO CURRENT ADDR REG
3960 023616 017777 155270 155204 MOV @RSBA,@DISPLAY ;SHOW IN LIGHTS
3961 023624 000770 BR 1$
3962
3963 ;DISK ADDRESS REGISTER
3964 023626 012706 000500 SRSDA: MOV #500,SP ;INIT THE STACK
3965 023632 104416 1$: KBDIN ;CTW
3966 023634 017777 155166 155252 MOV @SWR,@RSDA ;MOV SWR INTO DISK ADDR REG
3967 023642 017777 155246 155160 MOV @RSDA,@DISPLAY ;SHOW IN LIGHTS
3968 023650 000770 BR 1$
3969
3970 ;DRIVE STATUS REGISTER
3971 023652 012706 000500 SRSDS: MOV #500,SP ;INIT THE STACK
3972 023656 104416 1$: KBDIN ;CTW
3973 023660 017777 155142 155230 MOV @SWR,@RSDS ;MOV SWR INTO DRIVE STATUS
3974 023666 017777 155224 155134 MOV @RSDS,@DISPLAY ;SHOW IN LIGHTS
3975 023674 000770 BR 1$
3976
3977 ;DRIVE ERROR REGISTER
3978 023676 012706 000500 SRSER: MOV #500,SP ;INIT THE STACK
3979 023702 104416 1$: KBDIN ;CTW
3980 023704 017777 155116 155206 MOV @SWR,@RSER ;LOAD ER REG
3981 023712 017777 155202 155110 MOV @RSER,@DISPLAY ;DISPLAY IT IN LIGHTS
3982 023720 000770 BR 1$ ;LOOP
3983
3984 ;WATCH LOOK AHEAD REGISTER
3985 023722 017777 155176 155100 SRSLA: MOV @RSLA,@DISPLAY ;SHOW IN LIGHTS
3986 023730 000774 BR SRSLA
  
```

```

3987
3988 023732 012706 000500      ;RSCS2 REGISTER
3989 023736 104416      SRCS2: MOV #500,SP      ;INIT THE STACK
3990 023740 017777 155062 155140 1$: KBDIN      ;CTW
3991 023746 017777 155134 155054  MOV @SWR,@RSCS2      ;LOAD CS2
3992 023754 000770  MOV @RSCS2,@DISPLAY ;DISPLAY IT
3993  BR 1$
3994
3995 023756 012706 000500      ;RSAS REGISTER
3996 023762 104416      SRAS: MOV #500,SP      ;INIT THE STACK
3997 023764 017777 155036 155130 1$: KBDIN      ;CTW
3998 023772 017777 155124 155030  MOV @SWR,@RSAS      ;LOAD RSAS
3999 024000 000770  MOV @RSAS,@DISPLAY ;DISPLAY IT
4000  BR 1$
4001
4002 024002 012706 000500      ;RSMR REGISTER
4003 024006 104416      RSMRR: MOV #500,SP      ;INIT THE STACK
4004 024010 017777 155012 155112 1$: KBDIN      ;CTW
4005 024016 017777 155106 155004  MOV @SWR,@RSMR      ;LOAD RSMR
4006 024024 000770  MOV @RSMR,@DISPLAY ;DISPLAY IT
4007  BR 1$
4008
4009 024026 012706 000500      ;DISK CONTROL STATUS REGISTER
4010 024032 104416      SRSCS1: MOV #500,SP      ;INIT THE STACK
4011 024034 012737 000340 177776 1$: KBDIN      ;CTW
4012 024042 012777 177777 155040  MOV #340,@#PS      ;LOCK UP INTERRUPTS
4013 024050 013777 001102 155034  MOV #177777,@RSWC  ;SET WORD COUNT -1 WORD
4014 024056 017777 154744 155020  MOV @#OBUFSV,@RSBA ;SET UP CURRENT ADDRESS
4015 024064 032777 000001 155012  MOV @SWR,@RSCS1    ;MOV SWR INTO CONTROL REG
4016 024072 001757  BIT #BIT0,@RSCS1    ;IS FUNCTION BITS SET
4017 024074 105777 155004 2$: BEQ 1$           ;FUNCTION BITS NOT SET
4018 024100 100375  TSTB @RSCS1        ;TEST FOR DISK READY
4019 024102 000753  BPL 2$            ;DISK STILL NOT READY
                     BR 1$           ;DISK NOT BUSY SECT NEW CR
    
```

```

4020 :THIS ROUTINE GIVES THE OPERATOR THE ABILITY TO
4021 :SELECT DA, WC, UNIT # AND DESIRED PATTERN. PATTERN = NUMBER TYPED
4022 :WITH SW12 SET THE PROGRAM WILL LOOP ON A READ WITH LOC OUTBUF+2 AS
4023 :THE BA ADDR. WITH BIT12 0 IN THE SWR THE PROGRAM
4024 :WILL WRITE WITH OUTBUF AS THE BA ADDR. BAI IS ALWAYS SET
4025 :SWITCHES 0 TO 11 WILL DETERMINE THE DA
4026 :EXAMPLE:      TYPE UNIT # 5
4027 :              TYPE POSITIVE (OCTAL) WC 64
4028 :              TYPE PATTERN DESIRED 1252525
4029
4030 024104 012706 000500      TKSEL:  MOV      #500,SP      ;SET STACK
4031 024110 000240              NOP
4032 024112 104402 024116      TYPE      ..+2      ;.ASCIZ <15><12>'TYPE UNIT # ''
4033 024136 104410              RDOCT
4034 024140 012637 001152      MOV      (6)+,UNNUM
4035 024144 104402 024150      TYPE      ..+2      ;.ASCIZ <15><12>'TYPE POSITIVE (OCTAL) WC ''
4036 024204 104410              RDOCT
4037 024206 012637 001174      MOV      (6)+,WORK
4038 024212 005137 001174      COM      WORK
4039 024216 104402 024222      TYPE      ..+2      ;.ASCIZ <15><12>'TYPE PATTERN DESIRED ''
4040 024252 104410              RDOCT
4041 024254 012637 027454      MOV      (6)+,OUTBUF
4042 024260 042737 000001 027214  BIC      #BIT0,SWI      ;CLEAR THE BEENHEREBIT
4043 024266 104420              SUSWR
4044 024270 017737 154532 001200  TK1:    MOV      @SWR,WORK2      ;INIT SWITCHLESS
4045 024276 104414              TK2:    CLRDK      ;SAVE SWR
4046 024300 052777 000010 154600      BIS      #BIT3,@RSCS2      ;CLEAR ALL RS REG
4047 024306 017737 154514 001176      MOV      @SWR,WORK1      ;SET BAI
4048 024314 042737 170000 001176      BIC      #170000,WORK1      ;GET SWR FOR DSK ADDR
4049 024322 013777 001176 154564      TKKS:   MOV      WORK1,@RSDA      ;CLEAR UNIT #
4050 024330 013777 001174 154552      MOV      WORK,@RSWC      ;LOAD THE DA
4051 024336 032777 010000 154462      BIT      #BIT12,@SWR      ;LOAD WORD COUNT
4052 024344 001412              BEQ      WTE      ;READ?
4053 024346 012777 027456 154536      MOV      #OUTBUF+2,@RSBA      ;NO
4054 024354 012777 000071 154522      MOV      #71,@RSCS1      ;LOAD CURRENT ADDRESS
4055 024362 105777 154516      WT:    TSTB      @RSCS1      ;GO AND READ
4056 024366 100375              BPL      #-4      ;TEST FOR READY
4057 024370 000407              BR      SWRCHG
4058 024372 012777 027454 154512      WTE:    MOV      #OUTBUF,@RSBA
4059 024400 012777 000061 154476      MOV      #61,@RSCS1
4060 024406 000765              BR      WT
4061 024410 017777 154470 154412      SWRCHG: MOV      @RSCS1,@DISPLAY      ;DISPLAY CS1
4062 024416 005777 154462      TST      @RSCS1      ;ANY ERRORS?
4063 024422 100001              BPL      1$      ;NO
4064 024424 104014              HLT      !DA!WC
4065 024426 104416              1$:    KBDIN      ;CHECK FOR NEW VALUE
4066 024430 027737 154372 001200      CMP      @SWR,WORK2      ;DID SWR CHANGE?
4067 024436 001314              BNE      TK1      ;YES
4068 024440 000716              BR      TK2      ;NO

```

```

4069
4070
4071
4072 024442 104400
4073
4074 024444
4075 024444 104402 024450
4076 024510 022737 000176 001026
4077 024516 001402
4078 024520 000000
4079 024522 000401
4080 024524 104422
4081 024526 017737 154274 001152
4082 024534 104414
4083 024536 005037 027454
4084 024542 012777 027454 154342
4085 024550 012777 007700 154336
4086 024556 012777 177777 154324
4087 024564 012777 000061 154312
4088 024572 105777 154306
4089 024576 100375
4090 024600 104402 024604
4091 024652 000000
4092 024654 012777 000000 154232
4093 024662 022777 014600 154226
4094 024670 001401
4095 024672 104044
4096 024674 012777 000100 154212
4097 024702 022777 010600 154206
4098 024710 001401
4099 024712 104044
4100 024714 104402 024720
4101 024754 000000
4102 024756 022777 014600 154132
4103 024764 001401
4104 024766 104044
4105 024770 012777 000300 154116
4106 024776 022777 010600 154112
4107 025004 001401
4108 025006 104044
4109 025010 104402 025014
4110 025050 000000
4111 025052 022777 014600 154036
4112 025060 001401
4113 025062 104044
4114 025064 012777 000700 154022
4115 025072 022777 010600 154016
4116 025100 001401
4117 025102 104044
4118 025104 104402 025110
4119 025144 000000

```

```

*****
:TEST 122 WRITE LOCK TEST
*****
TST122: SCOPE
WRTLCK:
TYPE ..+2 ;.ASCIZ <15><12>'LOAD SW WITH UNIT # AND CONT''
CMP #SWREG,SWR
BEQ 7$ ;CHECK IF SWITCHLESS CPU
HALT
BR 8$ ;GO AROUND TRAP CALL
CNTLU ;GET SWREG VALUE
7$: MOV @SWR,UNNUM ;GET UNIT #
8$: CLRDK ;CLEAR ALL REG
CLR OUTBUF ;PUT A 0 INTO DATA BUFFER
MOV #OUTBUF,@RSBA ;SETUP REG TO
MOV #7700,@RSDA ;TO A WRITE
MOV #-1,@RSWC
MOV #61,@RSCS1
6$: TSTB @RSCS1 ;WAIT FOR DONE
BPL 6$
TYPE ..+2 ;.ASCIZ <15><12>'SET WRITE LOCK ENABLE AND CONTINUE''
HALT
MOV #0,@RSDA
1$: CMP #14600,@RSDS
BEQ .+4
HLT !DS!DA ;DS SHOULD=14600
MOV #100,@RSDA
CMP #10600,@RSDS
BEQ .+4
HLT !DS!DA ;DS SHOULD=10600
TYPE ..+2 ;.ASCIZ <15><12>'SET WRT LOC SW 0 AND CONT''
HALT
2$: CMP #14600,@RSDS
BEQ .+4
HLT !DS!DA ;DS SHOULD=14600
MOV #300,@RSDA
CMP #10600,@RSDS
BEQ .+4
HLT !DS!DA ;DS SHOULD=10600
TYPE ..+2 ;.ASCIZ <15><12>'SET WRT LOC SW 1 AND CONT''
HALT
3$: CMP #14600,@RSDS
BEQ .+4
HLT !DS!DA ;DS SHOULD=14600
MOV #700,@RSDA
CMP #10600,@RSDS
BEQ .+4
HLT !DS!DA ;DS SHOULD=10600
TYPE ..+2 ;.ASCIZ <15><12>'SET WRT LCK SW 2 AND CONT''
HALT

```

4120	025146	022777	014600	153742	4\$:	CMP	#14600,@RSDS	
4121	025154	001401				BEQ	+.4	
4122	025156	104044				HLT	!DS!DA	:DS SHOULD=14600
4123	025160	012777	001700	153726		MOV	#1700,@RSDA	
4124	025166	022777	010600	153722		CMP	#10600,@RSDS	
4125	025174	001401				BEQ	+.4	
4126	025176	104044				HLT	!DS!DA	:DS SHOULD=10600
4127	025200	104402	025204			TYPE	..+2	:.ASCIZ <15><12>'SET WRT LCK SW 3 AND CONT''
4128	025240	000000			HALT:	HALT		
4129	025242	022777	014600	153646		CMP	#14600,@RSDS	
4130	025250	001401				BEQ	+.4	
4131	025252	104044				HLT	!DS!DA	:DS SHOULD=14600
4132	025254	012777	003700	153632		MOV	#3700,@RSDA	
4133	025262	022777	010600	153626		CMP	#10600,@RSDS	
4134	025270	001401				BEQ	+.4	
4135	025272	104044				HLT	!DS!DA	:DS SHOULD=10600
4136	025274	104402	025300			TYPE	..+2	:.ASCIZ <15><12>'SET WRT LCK SW 4 AND CONT''
4137	025334	000000				HALT		
4138	025336	022777	014600	153552		CMP	#14600,@RSDS	
4139	025344	001401				BEQ	+.4	
4140	025346	104044				HLT	!DS!DA	:DS SHOULD=14600
4141	025350	012777	006000	153536		MOV	#6000,@RSDA	
4142	025356	022777	010600	153532		CMP	#10600,@RSDS	
4143	025364	001401				BEQ	+.4	
4144	025366	104044				HLT	!DS!DA	:DS SHOULD=10600
4145	025370	104402	025374			TYPE	..+2	:.ASCIZ <15><12>'SET WRT LCK SW 5 AND CONT''
4146	025430	000000				HALT		
4147	025432	022777	014600	153456		CMP	#14600,@RSDS	
4148	025440	001401				BEQ	+.4	
4149	025442	104044				HLT	!DS!DA	:DS SHOULD=14600
4150	025444	012777	007700	153442		MOV	#7700,@RSDA	
4151	025452	022777	014600	153436		CMP	#14600,@RSDS	
4152	025460	001401				BEQ	+.4	
4153	025462	104044				HLT	!DS!DA	:DS SHOULD=14600
4154	025464	012737	177777	027454		MOV	#-1,OUTBUF	:PUT A 1 INTO DATA BUFFER
4155	025472	012777	027454	153412		MOV	#OUTBUF,@RSBA	:SETUP REG TO
4156	025500	012777	007700	153406		MOV	#7700,@RSDA	:TO A WRITE
4157	025506	012777	177777	153374		MOV	#-1,@RSWC	
4158	025514	012777	000061	153362		MOV	#61,@RSCS1	:TRY TO WRITE
4159	025522	105777	153356		7\$:	TSTB	@RSCS1	:WAIT FOR DONE
4160	025526	100375				BPL	7\$	
4161	025530	105777	153350		5\$:	TSTB	@RSCS1	:WAIT FOR READY
4162	025534	100375				BPL	5\$	
4163	025536	022777	154600	153352		CMP	#154600,@RSDS	
4164	025544	001401				BEQ	+.4	
4165	025546	104044				HLT	!DS!DA	:DS SHOULD=154600

```

4166 025550 022777 004000 153342      CMP      #4000,@RSER
4167 025556 001401                    BEQ      .+4
4168 025560 104044                    HLT      !DS!DA      ;ER SHOULD=4000
4169 025562 104414                    CLRDK   ;CLEAR ALL REG
4170 025564 022777 014600 153324      CMP      #14600,@RSDS ;DID WLE CLEAR?
4171 025572 001401                    BEQ      .+4      ;NO
4172 025574 104040                    HLT      !DS      ;A CLEAR SHOULD NOT CLEAR WLE
4173 025576 005037 027454            CLR      OUTBUF    ;PUT A 0 INTO DATA BUFFER
4174 025602 012777 027454 153302      MOV      #OUTBUF,@RSBA ;SETUP REG TO
4175 025610 012777 007700 153276      MOV      #7700,@RSDA ;TO A WRITE
4176 025616 012777 177777 153264      MOV      #-1,@RSWC
4177 025624 012777 000071 153252      MOV      #71,@RSCS1 ;DO A READ TO SEE IF DISK DID
4178 025632 105777 153246      8$: TSTB   @RSCS1    ;ACTUALLY GET WRITTEN ON TO
4179 025636 100375                    BPL     8$        ;WAIT FOR DONE
4180 025640 005737 027454            TST     OUTBUF    ;IS DATA STILL 0 ON THE DSK?
4181 025644 001401                    BEQ     .+4      ;YES
4182 025646 104040                    HLT     !DS      ;NO DSK DID GET WRITTEN ONTO WITH WLE SET
4183 025650 104402 025654            TYPE   ..+2     ;.ASCIZ <15><12>'RESET ALL W/L SWITCHES''
4184 025706 000000                    HALT
4185 025710 022777 010600 153200      CMP      #10600,@RSDS ;DID WLE CLEAR
4186 025716 001401                    BEQ     .+4      ;YES
4187 025720 104040                    HLT     !DS      ;NO
4188
4189                                     ;ROUTINE FOR FINDING MEMORY ON 'B' PORT
4190 025722 012737 026006 000004      FINDM: MOV     #MAXREF,4 ;SET UP I/O BUS TRAP
4191 025730 012737 000340 000006      MOV     #340,6      ;SET PS
4192 025736 032777 010000 153062      BIT     #BIT12,@SWR ;INHIBIT OBUFSV FROM CHANGING?
4193 025744 001003                    BNE     EXREF      ;YES
4194 025746 062737 020000 001102      ADD     #20000,OBUFSV ;ADD 4 K
4195 025754 005777 153122      EXREF: TST     @OBUFSV ;LEGAL LOC ? IF NO TRAPS
4196 025760 012737 000006 000004      MOV     #6,4        ;RESTORE I/O BUS TRAP
4197 025766 005037 000006            CLR     6
4198 025772 022737 177446 001102      CMP     #177446,OBUFSV ;TEST FOR GREATER THEN 28K
4199 026000 103403                    BLO     MAXRF1     ;YES
4200 026002 000137 007032            JMP     WRTSTB     ;RETRY WRITING
4201 026006 022626                    MAXREF: CMP     (6)+,(6)+ ;CLEAR STACK
4202 026010 012737 000006 000004      MAXRF1: MOV     #6,4 ;RESTORE I/O BUS TRAP
4203 026016 005037 000006            CLR     6
4204 026022 012737 027454 001102      MOV     #OUTBUF,OBUFSV ;RESTORE ORIGINAL VALUE
4205 026030 032777 020000 152770      BIT     #BIT13,@SWR ;INHIBIT TYPEOUT?
4206 026036 001031                    BNE     1$        ;YES
4207 026040 032737 020000 001160      BIT     #BIT13,ONCEE ;DID WE TYPE THIS YET?
4208 026046 001025                    BNE     1$        ;YES
4209 026050 104402 026054            TYPE   ..+2     ;.ASCIZ <15><12>'COULD NOT FIND MEMORY ON -B- PORT'<15><
4210
4211                                     .EVEN
4211 026122 052737 020000 001160      1$: BIS     #BIT13,ONCEE ;SET TYPE FLAG
4212 026130 104400                    SCOPE  ;UPDATE
4213 026132 104400                    SCOPE  ;TEST NUMBERS
4214 026134 000137 007622            JMP     NXI:      ;CONT TESTING
4215
4216                                     ;CLEAR ALL DISK REGISTERS
4217 026140 012777 000040 152740      .CLRDK: MOV     #40,@RSCS2 ;CLEAR ALL DSK REG
4218 026146 013777 001152 152732      MOV     UNNUM,@RSCS2 ;GET UNIT NUMBER
4219 026154 000002                    RTI
    
```

```

4220 ;ERROR TYPTXTOUT ROUTINE
4221
4222 026156 005737 022644 RSREG: TST .HLTCT ;SHOULD WE TYPTXT GOOD AND BAD
4223 026162 001021 BNE 8$ ;NO
4224 026164 104402 026170 TYPE ..+2 ;.ASCIZ " BAD="
4225 026176 010046 MOV BAD,-(6) ;PUT BAD ON STACK
4226 026200 104404 TYPEO ;TYPE STACK IN OCTAL
4227 026202 104402 026206 TYPE ..+2 ;.ASCIZ " GOOD="
4228 026216 010146 MOV GOOD,-(6) ;PUT GOOD ON STACK
4229 026220 104404 TYPEO ;TYPE STACK IN OCTAL
4230 026222 000137 026660 JMP PTDONE ;GET OUT
4231 026226
4232 026226 104402 026232 8$: TYPE ..+2 ;.ASCIZ " CS1="
4233 026240 017746 152640 MOV @RSCS1,-(6) ;PUT @RSCS1 ON STACK
4234 026244 104404 TYPEO ;TYPE STACK IN OCTAL
4235 026246
4236 026246 104402 026252 1$: TYPE ..+2 ;.ASCIZ " ER="
4237 026260 017746 152634 MOV @RSER,-(6) ;PUT @RSER ON STACK
4238 026264 104404 TYPEO ;TYPE STACK IN OCTAL
4239 026266
4240 026266 104402 026272 2$: TYPE ..+2 ;.ASCIZ " CS2="
4241 026300 017746 152602 MOV @RSCS2,-(6) ;PUT @RSCS2 ON STACK
4242 026304 104404 TYPEO ;TYPE STACK IN OCTAL
4243 026306 032737 000200 022644 BIT #200,.HLTCT ;TYPTXT SECOND SET ?
4244 026314 001076 BNE SEEC ;YES
4245 026316 032737 000100 022644 BIT #AS,.HLTCT ;TYPTXT ER ?
4246 026324 001410 BEQ 3$ ;NO
4247 026326 104402 026332 TYPE ..+2 ;.ASCIZ " AS="
4248 026340 017746 152556 MOV @RSAS,-(6) ;PUT @RSAS ON STACK
4249 026344 104404 TYPEO ;TYPE STACK IN OCTAL
4250 026346 032737 000020 022644 3$: BIT #BA,.HLTCT ;TYPTXT BUS ASSRESS
4251 026354 001410 BEQ 4$ ;NO
4252 026356 104402 026362 TYPE ..+2 ;.ASCIZ " BA="
4253 026370 017746 152516 MOV @RSBA,-(6) ;PUT @RSBA ON STACK
4254 026374 104404 TYPEO ;TYPE STACK IN OCTAL
4255 026376 032737 000004 022644 4$: BIT #DA,.HLTCT ;TYPTXT DA ?
4256 026404 001410 BEQ 5$ ;NO
4257 026406 104402 026412 TYPE ..+2 ;.ASCIZ " DA="
4258 026420 017746 152470 MOV @RSDA,-(6) ;PUT @RSDA ON STACK
4259 026424 104404 TYPEO ;TYPE STACK IN OCTAL
4260 026426 032737 000010 022644 5$: BIT #WC,.HLTCT ;TYPTXT WC?
4261 026434 001410 BEQ 6$ ;NO
4262 026436 104402 026442 TYPE ..+2 ;.ASCIZ " WC="
4263 026450 017746 152434 MOV @RSWC,-(6) ;PUT @RSWC ON STACK
4264 026454 104404 TYPEO ;TYPE STACK IN OCTAL

```



4265	026456	032737	000040	022644	6\$:	BIT	#DS,.HLTCT	:DRIVE STATUS
4266	026464	001475				BEQ	PTDONE	:NO
4267	026466	104402	026472			TYPE	..+2	:.ASCIZ '' DS=''
4268	026500	017746	152412			MOV	@RSD!,-(6)	:PUT @RSDS ON STACK
4269	026504	104404				TYPEO		:TYPE STACK IN OCTAL
4270	026506	000137	026660			JMP	PTDONE	:GET OUT
4271	026512	042737	000200	022644	SEEC:	BIC	#200,.HLTCT	:CLEAR COMMON BIT
4272	026520	032737	000240	022644		BIT	#DT,.HLTCT	:TYPTXT DRIVE TYPE?
4273	026526	001410				BEQ	9\$	:NO
4274	026530	104402	026534			TYPE	..+2	:.ASCIZ '' DT=''
4275	026542	017746	152364			MOV	@RSDT,-(6)	:PUT @RSDT ON STACK
4276	026546	104404				TYPEO		:TYPE STACK IN OCTAL
4277	026550	032737	000210	022644	9\$:	BIT	#DB,.HLTCT	:TYPTXT DATA BUFFER
4278	026556	001410				BEQ	10\$	:NO
4279	026560	104402	026564			TYPE	..+2	:.ASCIZ '' DB=''
4280	026572	017746	152330			MOV	@RSDB,-(6)	:PUT @RSDB ON STACK
4281	026576	104404				TYPEO		:TYPE STACK IN OCTAL
4282	026600	032737	000220	022644	10\$:	BIT	#MR,.HLTCT	:TYPTXT MN?
4283	026606	001410				BEQ	11\$	:NO
4284	026610	104402	026614			TYPE	..+2	:.ASCIZ '' MR=''
4285	026622	017746	152302			MOV	@RSMR,-(6)	:PUT @RSMR ON STACK
4286	026626	104404				TYPEO		:TYPE STACK IN OCTAL
4287	026630	032737	000204	022644	11\$:	BIT	#LA,.HLTCT	:TYPTXT LA?
4288	026636	001410				BEQ	PTDONE	:NO
4289	026640	104402	026644			TYPE	..+2	:.ASCIZ '' LA=''
4290	026652	017746	152246			MOV	@RSLA,-(6)	:PUT @RSLA ON STACK
4291	026656	104404				TYPEO		:TYPE STACK IN OCTAL
4292	026660	052737	100000	001160	PTDONE:	BIS	#BIT15,ONCEE	:SET FORND ERROR FLAG
4293	026666	000207				RTS	PC	
4294								
4295	026670	005037	001174		WAITRY:	CLR	WORK	:CLEAR COUNTER
4296	026674	105777	152204		1\$:	TSTB	@RSCS1	:TEST READY
4297	026700	100406				BMI	2\$	:OK CONT
4298	026702	005237	001174			INC	WORK	:UPDATE COUNTER
4299	026706	005737	001174			TST	WORK	:DONE YET?
4300	026712	001403				BEQ	3\$	:READY DID NOT COME UP
4301	026714	000767				BR	1\$	:CONTINUE WAITING
4302	026716	062716	000002		2\$:	ADD	#2,(SP)	:UPDATE RETURN PC
4303	026722	000207			3\$:	RTS	PC	:RETURN

```

4304 ;RANDOM DATA GENERATOR SUBROUTINE
4305
4306 026724 013737 027066 027072 RANDOM: MOV LONUM,LOSAV
4307 026732 013737 027070 027074 MOV HINUM,HISAV
4308 026740 013700 027066 RAND1: MOV LONUM,R0 ;SET UP R0 WITH 5 DIGITS LOW
4309 026744 013704 027070 MOV HINUM,R4 ;SET UP R1 WITH 5 DIGITS HIGH
4310 026750 012703 000007 MOV #7,R3 ;SET UP SHIFT COUNT
4311 026754 005002 CLR R2 ;CLEAR R2
4312 026756 006300 SHIFT: ASL R0 ;SHIFT R0 LEFT AND
4313 026760 006104 ROL R4 ;ROTATE CARRY INTO LSB OF R1 INTO
4314 026762 006102 ROL R2 ;ROTATE CARRY OUT OF R1 INTO R2
4315 026764 005303 DEC R3 ;DECREMENT R3
4316 026766 001373 BNE SHIFT ;CONTINUE SHIFT LOOP
4317 026770 063700 027066 ADD LONUM,R0 ;ADDN IN NUMBER TO MAKE X 129
4318 026774 005504 ADC R4 ;PROPOGATE CARRY
4319 026776 063704 027070 ADD HINUM,R4 ;ADDN IN NUMBER TO MAKE X 129
4320 027002 005502 ADC R2 ;PROPOGATE CARRY
4321 027004 062700 001057 ADD #1057,R0 ;ADDN LOW CONSTANT
4322 027010 005504 ADC R4 ;PROPOGATE CARRIES
4323 027012 005502 ADC R2 ;PROPOGATE AGAIN
4324 027014 062704 047401 ADD #47401,R4 ;ADDN HIGH CONSTANT
4325 027020 005502 ADC R2 ;PROPOGATE CARRY
4326 027022 062702 000006 ADD #6,R2 ;ADDN HIGHEST CONSTANT
4327 027026 062700 000002 ADD #2,R0 ;REPRIME R0 WITH HIGH DIGIT
4328 027032 005504 ADC R4 ;PROPOGATE CARRY
4329 027034 010037 027066 MOV R0,LONUM ;PUT R0 BACK IN LONUM
4330 027040 010021 MOV R0,(R1)+ ;LOAD WC
4331 027042 005337 001174 DEC WORK
4332 027046 001406 BEQ EXGEN
4333 027050 010437 027070 MOV R4,HINUM ;PUT R1 BACK IN HINUM
4334 027054 010421 MOV R4,(1)+ ;HOLD HINUM FOR PROGRAM
4335 027056 005337 001174 DEC WORK
4336 027062 001326 BNE RAND1
4337 027064 000205 EXGEN: RTS ;RETURN TO PROGRAM
4338 027066 000000 LONUM: 0
4339 027070 000000 HINUM: 0
4340 027072 000000 LOSAV: 0
4341 027074 000000 HISAV: 0
4342 027076 RANEND:
4343
4344 027076 032737 000001 027214 .SUSWR: BIT #BIT0,SWI
4345 027104 001037 BNE XXX
4346 027106 013746 000006 MOV 6,-(SP) ;SAVE 6 ON STACK
4347 027112 013746 000004 MOV 4,-(SP) ;SAVE 4 ON STACK
4348 027116 012737 027136 000004 MOV #1$,4 ;SET UP TRAP ADDRESS
4349 027124 022777 177777 151674 CMP #-1,@SWR ;TEST 177570
4350 027132 001402 BEQ 2$ ;FAKE OUT
4351 027134 000407 BR 3$ ;HARDWARE AVAILABLE
4352 027136 022626 1$: CMP (SF)+,(SF)+ ;ADJUST STACK
4353 027140 012737 000176 001026 2$: MOV #SWREG,SWR ;SET UP SOFTWARE REGISTERS
4354 027146 012737 000174 001030 MOV #DISPREG,DISPLAY
4355 027154 022737 000176 001026 3$: CMP #SWREG,SWR ;1ST TIME THRU?
4356 027162 001004 BNE 4$ ;NO CHANGE STILL 177570
4357 027164 005737 000042 TST 42 ;ANY XXDP OR ACT
4358 027170 001001 BNE 4$ ;SWR=000000
4359 027172 104422 CNTLU ;GET INITIAL SETTINGS
    
```

```

4360 027174 012637 000004      4$:  MOV      (SP)+,4      ;REPLACE 4 FROM STACK
4361 027200 012637 000006      MOV      (SP)+,6      ;REPLACE 6 FROM STACK
4362 027204 052737 000001 027214 xxx:  BIS      #BIT0,SWI    ;SET THE BEENHEREBIT
4363 027212 000002      RTI                      ;ALL DONE
4364
4365 027214 000000      SWI:    0
4366
4367
4368
4369 027216 005737 000042      .KBDIN: TST      42      ;GOT XXDP OR ACT
4370 027222 001057      BNE OKT                      ;YES,GET OUT
4371 027224 022737 000176 001026  CMP      #SWREG,SWR      ;GOT SWITCH-LESS MACHINE?
4372 027232 001053      BNE OKT                      ;NO GET OUT
4373 027234 105777 151560      TSTB    @TKS              ;HAVE A CHARACTER
4374 027240 100050      BPL OKT                      ;NO GET OUT
4375 027242 017737 151554 027364  MOV      @TKB,MSG        ;
4376 027250 042737 177600 027364  BIC      #177600,MSG     ;STRIP OFF GARBAGE
4377 027256 122737 000007 027364  CMPB    #7,MSG          ;DO WE HAVE A ^G
4378 027264 001036      BNE OKT                      ;NO,GET OUT
4379 027266 104402 027272      TYPE    ..+2              ;.ASCIZ <15><12>'^G'
4380 027300
4381 027300 104402 027304      .CNTLU: TYPE    ..+2      ;.ASCIZ <15><12>'SWR= '
4382 027314 013746 000176      MOV      SWREG,-(6)      ;PUT SWREG ON STACK
4383 027320 104404      TYPE0                      ;TYPE STACK IN OCTAL
4384 027322 104402 027326      TYPE    ..+2              ;.ASCIZ '' NEW= ''
4385 027340 104410      RDOCT
4386 027342 012637 027364      MOV      (SP)+,MSG      ;GET NEW VALUE OFF STACK
4387 027346 005737 027366      TST      CTN              ;DID HE TYPE <CR> OF 000000?
4388 027352 001403      BEQ OKT                      ;DONT CHANGE IF <CR>
4389 027354 013737 027364 000176  MOV      .MSG,SWREG     ;CHANGE VALUE OF SWREG
4390 027362 000002      OKT:  RTI                  ;ALL DONE-EXIT
4391
4392 027364 000000      .MSG:    0
4393 027366 000000      CTN:    0
    
```

```
4394  
4395 ;TABLES FOR ILLEGAL FUNCTION TESTS  
4396  
4397 027370 000003 ILLTAB: 3  
4398 027372 000005 5  
4399 027374 000007 7  
4400 027376 000013 13  
4401 027400 000015 15  
4402 027402 000017 17  
4403 027404 000023 23  
4404 027406 000025 25  
4405 027410 000027 27  
4406 027412 000033 33  
4407 027414 000035 35  
4408 027416 000037 37  
4409 027420 000041 41  
4410 027422 000043 43  
4411 027424 000045 45  
4412 027426 000047 47  
4413 027430 000000 0  
4414  
4415 027432 000053 ILFTB2: 53  
4416 027434 000055 55  
4417 027436 000057 57  
4418 027440 000063 63  
4419 027442 000065 65  
4420 027444 000073 73  
4421 027446 000075 75  
4422 027450 000077 77  
4423 027452 000000 0  
4424 027454 000300 OUTBUF: .BLKW 300  
4425 030254 000300 INBUF: .BLKW 300  
4426 031054 000000 PARITY: 0 ;USED FOR PARITY TEST 67  
4427 000001 .END
```

ADDCF	012214	2357	2366#											
AROUND	001406	811	816#											
AS	= 000100	740#	961	1514	1524	1546	1577	1590	1612	1655	2080	2138	2297	2612
		2886	2926	3019	3026	3063	3084	3090	3556	3576	3582	3605	3609	3612
		4245												
ATA	= 100000	761#												
AOB1	001166	775#												
BA	= 000020	738#	931	989	995	1077	1081	1085	1107	1346	1350	1511	1543	1574
		1609	1675	2086	2141	2208	2223	3060	3123	4250				
		764#	1985	1998	2007	3110	3133	3167	3251					
BAI	= 000010	1982#												
BAITST	010050	615	633	795#	3641									
BEGIN	001202	642#	3636	3740										
BELL	= 000007	1342#												
BITBA	004444	1318#												
BITCS2	004330	1305#												
BITST	004256	1330#												
BITWC	004374	651#	878	896	1290	1676	1726	1779	1858	1911	2112	2180	4015	4042
BITO	= 000001	4344	4362											
		652#												
BIT1	= 000002	661#												
BIT10	= 002000	614	632	662#	806	1816								
BIT11	= 004000	663#	1814	2936	2939	4051	4192							
BIT12	= 010000	664#	834	860	882	2523	4205	4207	4211					
BIT13	= 020000	601	665#	2457	2482	2506	2509	2513	2534	2559	2583	2586	2591	
BIT14	= 040000	666#	837	862	871	1375	3551	4292						
BIT15	= 100000	653#												
BIT2	= 000004	654#	3520	3526	3588	4046								
BIT3	= 000010	655#	3073	3114	3139	3174								
BIT4	= 000020	656#												
BIT5	= 000040	657#	3183											
BIT6	= 000100	658#	2736											
BIT7	= 000200	659#												
BIT8	= 000400	660#	2356	2360										
BIT9	= 001000	2035#												
BLOCK	010502	778#	1923*	2526	2528*	2536*	2541*	2543*	2557	2588*				
BPORTT	001170	874	878#											
CHCKDV	001744	1019	1305	1351	1359	1384	1409	1455	1492	1520	1555	1586	1626	1665
CLRDK	= 104414	1716	1760	1803	1848	1891	1932	1982	1997	2006	2035	2105	2148	2167
		2227	2241	2263	2305	2310	2354	2413	2450	2496	2510	2527	2573	2587
		2597	2639	2663	2676	2694	2701	2719	2725	2750	2761	2835	2855	2873
		2890	2902	2945	2968	2980	3003	3012	3072	3094	3109	3132	3157	3166
		3193	3205	3228	3240	3300	3357	3406	3431	3453	3481	3504	3942#	4045
		4082	4169											
		3945#	4080	4359										
CNTLU	= 104422	2352#												
CSTST	012144	734#	926	984	1024	1028	1032	1040	1310	1314	1369	1373	1499	1531
CS1	= 000001	1562	1597	1652	1679	1681	1687	1723	1725	1731	1767	1769	1811	1813
		1824	1855	1857	1863	1899	1901	1910	1948	1951	1988	2021	2064	2086
		2123	2126	2141	2144	2147	2191	2194	2197	2208	2211	2223	2226	2489
		2505	2566	2582	2606	2609	2621	2650	2684	2709	2740	2795	2861	2889
		2958	2967	2993	3002	3037	3218	3227	3415	3421	3440	3470	3470	
		741#	918	980	1050	1060	1064	1069	1366	1378	1395	1420	1431	1443
CS2	= 000200	2028	2630	2923	2930									
CTN	027366	3862*	3879*	4387	4393#									
DA	= 000004	736#	937	1152	1156	1160	1181	1508	1540	1571	1606	1641	1684	1687









		2599	2640	2677	2702	2726	2772	2779	2796	2840	2857	2875	2955	2990
		3032	3058	3112	3121	3135	3169	3215	3242	3257	3269	3275	3437	3460
		3485	3534	3567	4013	4194*	4195	4198	4204*					
OKT	027362	4370	4372	4374	4378	4388	4390#							
ONCEE	001160	614*	632*	772#	804*	806	837*	862	871*	878	896*	2457	2482	2506
		2509*	2513*	2523	2534	2559	2583	2586*	2591*	2936	2939*	4207	4211*	4292*
ODUT	010600	2050	2090	2092#										
OR =	000200	752#	1429											
OUT	022046	3627#												
OUTBUF	027454	805	895	1253	1255	1269	1283	1493*	1525*	1556*	1591*	1666*	1717*	1747
		1761*	1924	1937*	2451*	2459*	2461*	2464*	2466*	2480	2511*	2538*	2590	2598*
		2771*	2778*	2821	3031*	3111*	3134*	3168*	3241*	3256*	3533*	4041*	4053	4058
		4083*	4084	4154*	4155	4173*	4174	4180	4204	4424#				
PARITY	031054	2396	2414	2440*	4426#									
PART	012334	2396#												
PARTST	011742	2304#												
PATST	016422	3072#												
PATTST	017062	3166#												
PCNT	001004	674#	3631*	3632*	3643									
PGE =	002000	753#	2626											
PGETST	013506	2597#												
PGTRAP	020612	3408	3418#											
PIP =	020000	758#												
PRTP	012512	2398	2432#											
PRTP1	012514	2408	2433#											
PS =	177776	640#	641	903*	967*	3410*	3434*	3435	3456*	3457	3479*	3547*	3619*	3625*
		4011*												
PSW =	177776	641#												
PTDONE	026660	4230	4266	4270	4288	4292#								
QES	020532	3406#												
QESDON	020626	3416	3422#											
QQ =	000001	819#	899#	962#	1012#	1033#	1042#	1071#	1087#	1099#	1108#	1124#	1138#	1145#
		1161#	1175#	1182#	1195#	1201#	1207#	1214#	1224#	1231#	1239#	1247#	1300#	1316#
		1328#	1340#	1352#	1379#	1406#	1452#	1487#	1515#	1550#	1581#	1619#	1660#	1711#
		1752#	1798#	1843#	1885#	1927#	1977#	2001#	2030#	2093#	2157#	2236#	2299#	2321#
		2329#	2347#	2391#	2441#	2514#	2592#	2634#	2671#	2696#	2720#	2754#	2850#	2866#
		2897#	2940#	2975#	3007#	3067#	3104#	3128#	3161#	3200#	3236#	3295#	3401#	3424#
		3447#	3473#	3495#	4069#									
RANDOM	026724	1254	4306#											
RAND1	026740	4308#	4336											
RANEND	027076	4342#												
RANNU	001150	768#												
RANTS	004036	1252#												
RDLIN =	104412	3860	3941#											
RDOCT =	104410	3940#	4033	4036	4040	4385								
RDTST	006472	1716#												
RDTSTB	007250	1848#												
RMRT1	013706	2639#												
RMRT2	014064	2676#												
RMRT3	014174	2701#												
RMRT4	014304	2725#												
RSAS	001122	698#	831	959	1512	1522	1544	1575	1588	1610	1653	2073	2075	2077*
		2078	2135	2295	2610	2734*	2924	3014	3023*	3024	3061	3085	3599	3615*
		3997*	3998	4248										
RSBA	001112	694#	906*	929	970*	987	1074*	1075	1078*	1079	1082*	1083	1092*	1093
		1103*	1104*	1105	1288	1293	1342*	1344	1347*	1348	1494*	1509	1526*	1541

		1557*	1572	1592*	1607	1667*	1673	1700	1718*	1741	1762*	1785	1806*	1837
		1850*	1873	1894*	1917	1938*	1967	1983*	1990	2009*	2037*	2084	2109*	2139
		2172*	2206	2221	2245*	2249	2270*	2275	2414*	2452*	2468*	2490	2529*	2545*
		2567	2599*	2640*	2677*	2702*	2726*	2772*	2779*	2796*	2831	2840*	2857*	2875*
		2955*	2990*	3032*	3058	3112*	3121	3135*	3169*	3215*	3242*	3257*	3275*	3321*
		3337*	3361*	3377*	3437*	3460*	3485*	3534*	3566	3959*	3960	4013*	4053*	4058*
		4084*	4155*	4174*	4253									
RSBAB	001146	708#	1343*											
RSCS1	001104	691#	905*	924	969*	982	1021*	1022	1025*	1026	1029*	1030	1036*	1037*
		1038	1306*	1308	1311*	1312	1367	1370*	1371	1496*	1497	1528*	1529	1559*
		1560	1578	1594*	1595	1616	1631*	1650	1669*	1676*	1677	1685	1720*	1721
		1729	1764*	1765	1776	1808*	1809	1822	1852*	1853	1861	1896*	1897	1908
		1949	1986*	2010*	2019	2022*	2046*	2062	2111*	2114	2117	2152	2179*	2182
		2185	2228	2246*	2247	2271*	2272	2288*	2289	2416*	2417	2425	2454*	2455
		2471*	2472	2487	2503	2531*	2532	2548*	2549	2564	2580	2601*	2602	2604*
		2607	2625*	2631	2647*	2660	2667	2679*	2680	691	2704*	2705	2707*	2716
		2733*	2747	2751	2774*	2775	2786*	2793	2798*	2799	2807	2841*	2842	2859*
		2877*	2878	2887	2921	2956*	2965	2991*	3000	3020	3028	3034*	3035	3038
		3040	3091	3095	3115*	3116	3118	3137*	3140	3142	3158	3171*	3172	3175
		3177	3194	3216*	3225	3252*	3253	3260*	3261	3285*	3286	3322*	3325	3327
		3331	3339*	3341	3343	3347	3363*	3364	3366	3369	3378*	3379	3381	3385
		3411*	3419	3438*	3461*	3468	3486*	3490	3538*	3544*	3550	3587*	3592*	3610
		3616*	4014*	4015	4017	4054*	4055	4059*	4061	4062	4087*	4088	4158*	4159
		4161	4177*	4178	4233	4296								
RSCS1B	001140	705#	1307*	2927*										
RSCS2	001106	692#	825*	829*	840*	904*	911*	916	919*	968*	975*	977*	978	981*
		1048	1051*	1052	1054	1057*	1058	1061*	1062	1065*	1066*	1067	1070*	1318*
		1319*	1323	1361	1374	1390	1417	1429	1441	1632	1696	1737	1772	1816
		1833	1869	1904	1963	1985*	1998	2007*	2015	2023	2127	2200	2213	2256
		2281	2419	2474	2499	2551	2576	2615	2627	2773*	2791*	2792*	2810	2911*
		2916	2928	2932	3043	3073*	3075	3098	3110*	3114*	3133*	3139*	3145	3167*
		3174*	3180	3251*	3259*	3263*	3290	3541*	3543*	3549*	3562	3606*	3990*	3991
		4046*	4217*	4218*	4241									
RSCS2B	001142	706#	1320*											
RSDA	001114	695#	907*	935	971*	993	1149*	1150	1153*	1154	1157*	1158	1167*	1168
		1177*	1178*	1179	1274	1278	1506	1538	1569	1604	1628*	1639	1682	1726
		1779	1819	1858	1911	1952	2036*	2045*	2145	2195	2366*	2368*	2369	2381
		2470*	2547*	2648*	2654	2777*	2858*	2862	2876*	2912*	2931	2954*	2989*	3055
		3206*	3255*	3274*	3336*	3376*	3483*	3536*	3574	3580	3584	3966*	3967	4049*
		4085*	4092*	4096*	4105*	4114*	4123*	4132*	4141*	4150*	4156*	4175*	4258	
RSDB	001126	700#	973*	1360	1385*	1386*	1397	1402	1412*	1425	1437*	1447	1458*	1464*
		1472	1480	4280										
RSDS	001116	696#	920	1532	1598	1691	1828	1958	2047	2051	2057	2059	2081	2124
		2192	2307	2314	2657	2688	2713	2736	2744	2804	2884	2962	2969	2997
		3004	3049	3151	3187	3222	3229	3607	3973*	3974	4093	4097	4102	4106
		4111	4115	4120	4124	4129	4133	4138	4142	4147	4151	4163	4170	4185
		4268												
RSDT	001132	702#	841	845	849	853	855	4275						
RSER	001120	697#	826*	908*	941	972*	999	1186*	1187	1190*	1191	1197*	1198	1203*
		1204	1209*	1210*	1211	1500	1521*	1547	1563	1587*	1613	1629*	1642	1706
		1994	2121	2149	2189	2231	2260	2264	2285	2292	2306*	2311	2622	2651
		2664	2682*	2685	2710	2741	2801	2836	2881	2891	2913	2959	2972	2994
		3013*	3064	3082	3154	3190	3197	3219	3980*	3981	4166	4237		
RSLA	001124	699#	2326	2334	2338	2375	3985	4290						
RSMR	001130	701#	909*	947	976*	1004	1218*	1219	1226*	1227*	1228	1233*	1234	1241*
		1242	1630*	1645	2040	2356	2360	2373	3074	3081*	4004*	4005	4285	







TYPEO = 104404	4252	4257	4262	4267	4274	4279	4284	4289	4379	4381	4384	4269	4276
TYPES = 104406	3749	3938#	4226	4229	4234	4238	4242	4249	4254	4259	4264		
UNCMP 001156	4281	4286	4291	4383									
UNITST 007640	868	885	3939#										
UNITSV 001154	771#	832*	838	857	873*	880*	1512	1522	1544	1575	1588	1610	1653
UNNUM 001152	2133	3015	3023	3061	3086								
	1932#												
	770#	831*	838	857*	2905	3512							
	769#	833*	840	867	875*	881*	884	904	919	968	981	1070	1318
	1321	1363	1389	1416	1634	1694	1735	1770	1831	1867	1902	1934	1941
	1961	2013	2024	2065	2128	2201	2214	2254	2279	2421	2475	2497	2552
	2574	2613	2792	2811	3045	3077	3099	3147	3182	3288	3514	3522	3549
	3560	4034*	4081*	4218									
UP = 000000	722#	3313	3314	3315	3316								
WAITRY 026670	1680	1724	1768	1812	1856	1900	1947	1987	2011	2605	2649	2683	2708
	2860	2957	2992	3217	3439	3487	4295#						
WC = 000010	737#	955	1011	1115	1119	1123	1144	1334	1338	1505	1537	1568	1603
	1658	1672	1690	1734	1784	1827	1866	1916	1957	2089	2144	2211	2226
	2621	3054	3126	3559	4064	4260							
WCE = 040000	755#												
WCEDNB 013446	2544	2586#											
WCEDNE 013062	2467	2509#											
WCEDON 013104	2462	2513#											
WCEDOS 013470	2525	2539	2590#										
WCETB 013144	2529#	2589											
WCETSB 013114	2523#												
WCETST 012552	2450#												
WCETT 012560	2452#	2512											
WORK 001174	791#	1219*	1220*	1221	1234*	1235*	1236	1242*	1243*	1244	1252*	1387*	1393*
	1645*	1646*	1647	1804*	1805*	1849*	1879*	1880	1892*	1893*	2039*	2042*	2065*
	2069	2071*	2169*	2178*	2198	2219*	2220*	2221	2243*	2244*	2245	2268*	2269*
	2270	2333*	2336*	2367*	2371*	2735*	2738*	2903*	2905	2908*	2931*	3266*	3268*
	3272*	3412*	3413*	3462*	3463*	3511*	3512	3518*	3598*	3599	3603*	3615	4037*
	4038*	4050	4295*	4298*	4299	4331*	4335*						
WORK1 001176	792#	2174*	2175*	2176	2377*	2379*	3507*	3520	3526*	3588	4047*	4048*	4049
WORK2 001200	793#	2764*	2766*	2769*	2770	2797	4044*	4066					
WRCKT 006660	1760#												
WRCKTB 007444	1891#												
WRTLCK 024444	611	4074#											
WRTST 006244	1665#												
WRTSTB 007032	1803#	4200											
WT 024362	4055#	4060											
WTDV 021426	3548#												
WTDV1 021776	3589	3614	3619#										
WTDV2 021656	3591	3594#	3617										
WTE 024372	4052	4058#											
WWP = 000004	776#	2402											
XXX 027204	4345	4362#											
ZERONE 004620	1384#												
\$ENDAD 022116	599	3639#											
\$END1 022126	3637	3641#											
. = 031056	597#	598#	600#	603#	607#	610#	613#	619#	631#	670#	685#	807	815
	836	864	887	888#	917	921	925	930	936	942	948	954	960
	979	983	988	994	1000	1005	1010	1023	1027	1031	1039	1049	1059
	1063	1068	1076	1080	1084	1095	1106	1114	1118	1122	1133	1143	1151
	1155	1159	1170	1180	1188	1192	1199	1205	1212	1222	1229	1237	1245



.TRAP	023512	800	3930#
.TRP	023530	3933	3934#
.TYPE	022154	3656#	3937
.TYPEB	022646	3768#	
.TYPEO	022656	3770#	3938
.TYPES	022666	3772#	3939





\$TYPED 1#  
\$SUMMR 1#

. ABS. 031056 000

ERRORS DETECTED: 0

C.BIN,C.LST/CRF/SOL/NL:TOC=CZRSBG.SML,CZRSBG.P11  
RUN-TIME: 10 15 1 SECONDS  
RUN-TIME RATIO: 79/28=2.8  
CORE USED: 22K (43 PAGES)