RH70/RS03

BASIC FUNCTIONAL DIAG

MD-11-DERSA-A

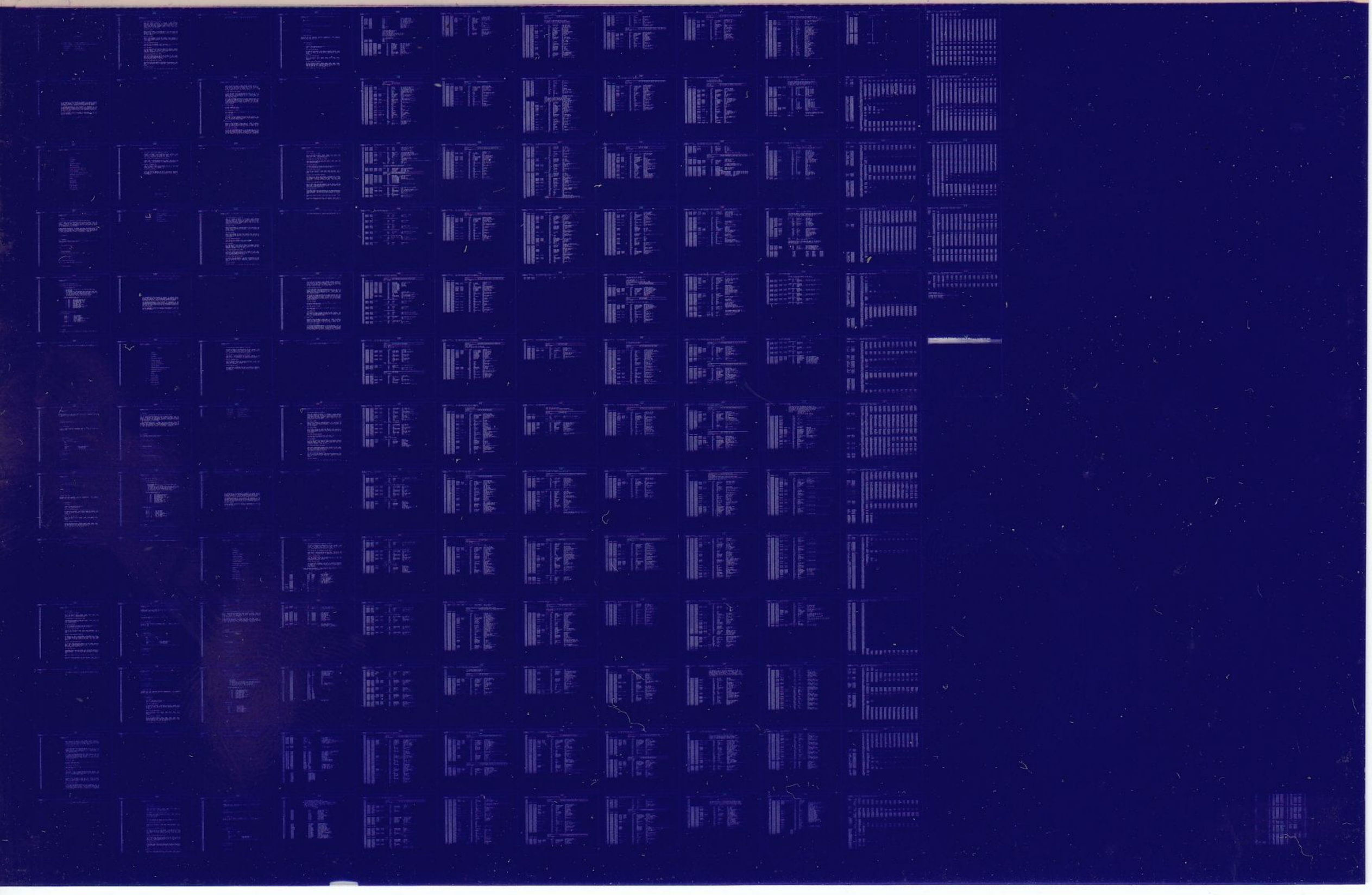
EP-DERSA-A-DL-A

FICHE 1 OF 1

NOV 1976

digital

MADE IN US



```
.REM%
```

IDENTIFICATION

PRODUCT CODE:	MAINDEC-11-DERSA-A-D
PRODUCT NAME:	RH11-RS03-RS04 BASIC FUNCTION DIAGNOSTIC
DATE CREATED:	MARCH-1975
MAINTAINER:	DIAGNOSTIC GROUP
AUTHORS:	STANLEY HARACKIEWICZ

TABLE OF CONTENTS

CONTENTS

DESCRIPTION

1. ABSTRACT

THIS DIAGNOSTIC WAS DESIGNED TO TEST RS03 AND RS04 DRIVES.

THIS IS A BASIC FUNCTION DIAGNOSTIC WHICH IS USED TO VERIFY THAT THE (RH11) CONTROLLER AND THE (RS03 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 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

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
   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 AND RS04 DISKS.

2. STARTING ADDRESSES FOR TESTING THE RH11-RS03/04 REGISTERS USING THE SWITCH REGISTER.

   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

SWITCH SETTINGS ARE:

```
SW<15> = 1 ..... HALT ON ERROR
SW<14> = 1 ..... LOOP ON TEST
SW<13> = 1 ..... INHIBIT TYPEOUTS
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

```
231                     SECTION.  IT  RECORDS  THE  STARTING ADDRESS OF EACH SUBTEST AS IT IS
232                     BEING ENTERED IN LOCATION "LAD".  IF A SCOPE LOOP  IS  REQUESTED,  THE
```

DESCRIPTION

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 INTERUPTS WILL HALT AT THE VECTOR + 2.

6. ERRORS

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 INITALLY 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

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

```
344
345                     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. 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.

8. TEST NO-OP FUNCTION

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

9. 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.

10. 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.

11. 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.

12. 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.

13. 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
                        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.

14. 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.

15. TEST PAR IN RSER

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

16. TEST DPR AND MOL IN RSDS

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

17. 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.

18. 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.

19. 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
                        WRITE CHECK FUNCTION TEST IS REPEATED.
```

20. 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.

21. 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.

22. 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.

23. TEST DISK ADDRESS REGISTER

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

24. 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.

25. 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.

26. 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.

27. TEST BAI IN RSCS2

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

518 IF IT INCREMENTED.

28. TEST NON-EXISTENT MEMORY ERROR BIT IN RSCS2

SET BITS 0 THROUGH 5 IN RHBAE FOR AN 22 BIT ADDRESS. MOV 173000 INTO RSBA. MOV -1000 INTO RSWC AND DO A WRITE FUNCTION. THE NEM BIT SHOULD SET AND SHOULD CAUSE TRE TO SET. CLEAR THESE BITS BY LOADING A 1 INTO TRE.

29. TEST FOR ZERO'S IN A PARTIALLY FILLED SECTOR

FIRST WRITE A COMPLETE SECTOR WITH ALL ONES. THEN DO A ONE WORD WRITE. THE REMAINING 63 WORDS SHOULD BE WRITTEN AS ZERO'S. NOW DO A WRITE CHECK TO COMPARE FOR THESE ZERO'S.

30. PRIORITY INTERRUPT TEST

HERE THE PROGRAM ENABLES THE INTERRUPT AND DOES A ONE WORD WRITE FUNCTION. THE PROGRAM SHOULD NOT TRAP UNTIL THE PROCESSOR IS DROPPED TO PRIORITY 4.

31. DYNAMIC FUNCTION TEST

WHILE ONE DRIVE IS READING, THE UNIT # IN RSCS2 IS MODIFIED. IF THERE IS ANOTHER DRIVE ON THE SYSTEM, A DRIVE SEARCH IS PERFORMED ON IT. THIS IS ALL DONE WHILE THE FIRST DRIVE IS STILL READING.

%
.REM%

IDENTIFICATION

```
575
576
577
578
579
580
581
582
583
584
```

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

PRODUCT NAME: RH11-RS03-RS04 BASIC FUNCTION DIAGNOSTIC

DATE CREATED: MARCH-1975

MAINTAINER: DIAGNOSTIC GROUP

AUTHORS: STANLEY HARACKIEWICZ

TABLE OF CONTENTS

CONTENTS

1. ABSTRACT

THIS DIAGNOSTIC WAS DESIGNED TO TEST RS03 AND RS04 DRIVES.

THIS IS A BASIC FUNCTION DIAGNOSTIC WHICH IS USED TO VERIFY THAT THE (RH11) CONTROLLER AND THE (RS03 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 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

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
   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 AND RS04 DISKS.

2. STARTING ADDRESSES FOR TESTING THE RH11-RS03/04 REGISTERS USING THE SWITCH REGISTER.

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

SWITCH SETTINGS ARE:

```
SW<15> = 1 ..... HALT ON ERROR
SW<14> = 1 ..... LOOP ON TEST
SW<13> = 1 ..... INHIBIT TYPEOUTS
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

```
779                     SECTION.  IT  RECORDS  THE  STARTING ADDRESS OF EACH SUBTEST AS IT IS
780                     BEING ENTERED IN LOCATION "LAD".  IF A SCOPE LOOP  IS  REQUESTED,  THE
```

DESCRIPTION

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 INTERUPTS WILL HALT AT THE VECTOR + 2.

6. ERRORS

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 INITALLY 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

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

```
892
893                     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. 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.

8. TEST NO-OP FUNCTION

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

9. 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.

10. 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.

11. 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.

12. 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.

13. 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
                        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.

14. 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.

15. TEST PAR IN RSER

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

16. TEST DPR AND MOL IN RSDS

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

17. 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.

18. 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.

19. 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

```
1008                                    A  1  IS  FLOATED THROUGH THE WORD IN THE BUS ADDRESS AND THE
1009                                    WRITE CHECK FUNCTION TEST IS REPEATED.
```

20. 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.

21. 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.

22. 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.

23. TEST DISK ADDRESS REGISTER

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

24. 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.

25. 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.

26. 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.

27. TEST BAI IN RSCS2

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

1066 IF IT INCREMENTED.

DESCRIPTION

28. TEST NON-EXISTENT MEMORY ERROR BIT IN RSCS2

SET BITS 0 THROUGH 5 IN RHBAE FOR AN 22 BIT ADDRESS. MOV 173000 INTO RSBA. MOV -1000 INTO RSWC AND DO A WRITE FUNCTION. THE NEM BIT SHOULD SET AND SHOULD CAUSE TRE TO SET. CLEAR THESE BITS BY LOADING A 1 INTO TRE.

29. TEST FOR ZERO'S IN A PARTIALLY FILLED SECTOR

FIRST WRITE A COMPLETE SECTOR WITH ALL ONES. THEN DO A ONE WORD WRITE. THE REMAINING 63 WORDS SHOULD BE WRITTEN AS ZERO'S. NOW DO A WRITE CHECK TO COMPARE FOR THESE ZERO'S.

30. PRIORITY INTERRUPT TEST

HERE THE PROGRAM ENABLES THE INTERRUPT AND DOES A ONE WORD WRITE FUNCTION. THE PROGRAM SHOULD NOT TRAP UNTIL THE PROCESSOR IS DROPPED TO PRIORITY 4.

31. DYNAMIC FUNCTION TEST

WHILE ONE DRIVE IS READING, THE UNIT # IN RSCS2 IS MODIFIED. IF THERE IS ANOTHER DRIVE ON THE SYSTEM, A DRIVE SEARCH IS PERFORMED ON IT. THIS IS ALL DONE WHILE THE FIRST DRIVE IS STILL READING.

%
.REM%

IDENTIFICATION

```
1123
1124
1125
1126
1127
1128
1129
1130
1131
1132
```

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

PRODUCT NAME: RH11-RS03-RS04 BASIC FUNCTION DIAGNOSTIC

DATE CREATED: MARCH-1975

MAINTAINER: DIAGNOSTIC GROUP

AUTHORS: STANLEY HARACKIEWICZ

CONTENTS

1. ABSTRACT

THIS DIAGNOSTIC WAS DESIGNED TO TEST RS03 AND RS04 DRIVES.

THIS IS A BASIC FUNCTION DIAGNOSTIC WHICH IS USED TO VERIFY THAT THE (RH11) CONTROLLER AND THE (RS03 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 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

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
   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 AND RS04 DISKS.

2. STARTING ADDRESSES FOR TESTING THE RH11-RS03/04 REGISTERS USING THE SWITCH REGISTER.

   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

SWITCH SETTINGS ARE:

```
SW<15> = 1 ..... HALT ON ERROR
SW<14> = 1 ..... LOOP ON TEST
SW<13> = 1 ..... INHIBIT TYPEOUTS
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

```
1327                    SECTION.  IT  RECORDS  THE  STARTING ADDRESS OF EACH SUBTEST AS IT IS
1328                    BEING ENTERED IN LOCATION "LAD".  IF A SCOPE LOOP  IS  REQUESTED,  THE
```

DESCRIPTION

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 INTERUPTS WILL HALT AT THE VECTOR + 2.

6. ERRORS

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 INITALLY 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

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

```
1440
1441                    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. 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.

8. TEST NO-OP FUNCTION

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

9. 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.

10. 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.

11. 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.

12. 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.

13. 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

```
1498                                    BLOCK SEARCH FOR SECTOR 0.  NOW WE KNOW THAT WE HAVE TIME  TO
1499                                    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 ILL OCCUR INDICATING SECTOR 0 WAS NOT FOUND. SC IN RSCS1 SHOULD ALSO SET. RSBA AND RSWC SHOULD NOT MOVE, THIS IS ALSO TESTED.

14. 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.

15. TEST PAR IN RSER

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

16. TEST DPR AND MOL IN RSDS

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

17. 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.

18. 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.

19. 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

```
1556                    A  1  IS  FLOATED THROUGH THE WORD IN THE BUS ADDRESS AND THE
1557                    WRITE CHECK FUNCTION TEST IS REPEATED.
```

20. 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.

21. 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.

22. 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.

23. TEST DISK ADDRESS REGISTER

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

24. 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.

25. 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.

26. 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.

27. TEST BAI IN RSCS2

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

1614 IF IT INCREMENTED.

28. TEST NON-EXISTENT MEMORY ERROR BIT IN RSCS2

SET BITS 0 THROUGH 5 IN RHBAE FOR AN 22 BIT ADDRESS. MOV 173000 INTO RSBA. MOV -1000 INTO RSWC AND DO A WRITE FUNCTION. THE NEM BIT SHOULD SET AND SHOULD CAUSE TRE TO SET. CLEAR THESE BITS BY LOADING A 1 INTO TRE.

29. TEST FOR ZERO'S IN A PARTIALLY FILLED SECTOR

FIRST WRITE A COMPLETE SECTOR WITH ALL ONES. THEN DO A ONE WORD WRITE. THE REMAINING 63 WORDS SHOULD BE WRITTEN AS ZERO'S. NOW DO A WRITE CHECK TO COMPARE FOR THESE ZERO'S.

30. PRIORITY INTERRUPT TEST

HERE THE PROGRAM ENABLES THE INTERRUPT AND DOES A ONE WORD WRITE FUNCTION. THE PROGRAM SHOULD NOT TRAP UNTIL THE PROCESSOR IS DROPPED TO PRIORITY 4.

31. DYNAMIC FUNCTION TEST

WHILE ONE DRIVE IS READING, THE UNIT # IN RSCS2 IS MODIFIED. IF THERE IS ANOTHER DRIVE ON THE SYSTEM, A DRIVE SEARCH IS PERFORMED ON IT. THIS IS ALL DONE WHILE THE FIRST DRIVE IS STILL READING.

```
                        %
                        .TITLE  MAINDEC-11-DERSA-A      RS11-RS03-RS04 BASIC FUNCTION DIAGNOSTIC
                        ;COPYRIGHT 1973,1974 DIGITAL EQUIPMENT CORP., MAYNARD, MASS.
                        ;PROGRAM BY STANLEY HARACKIEWICZ


                        ;           SWITCH                      USE
                        ;           ------                      ------------------
                                SW15=   100000          ;HALT ON ERROR
                                SW14=   40000           ;LOOP ON TEST
                                SW13=   20000           ;INHIBIT ERROR TYPEOUTS
                                SW12=   10000           ;
                                SW11=   4000            ;INHIBIT ITERATIONS
                                SW10=   2000            ;0 - BELL ON PASS COMPLETE
                                                        ;1 - BELL ON ERROR
                                SW9=    1000            ;LOOP ON ERROR
                                SW8=    400             ;LOOP ON TEST IN SW<7:0>
                        .=      0                       ;TRAP CATCHER FROM 0 - 776
                        .=      46                      ;HOOKS FOR ACT 11
                        $ENDAD
                        .=      52
                        BIT14
                        .=      200
                                JMP     @#STRT

                        .=      220
                                JMP     @#WRTLCK        ;WRITE LOCK TEST
```

```
1671
1672                    000230                  .=      230
1673    000230  042767  004000  000726  STRT:   BIC     #BIT11,ONCEE    ;CLEAR TKSEL TEST FLAG
1674    000236  000137  001206                  JMP     @#BEGIN         ;STARTING ADDRESS IS BEGIN
1675
1676                                    ;STARTING ADDRESSES  FOR SWITCH REGISTER TESTING OF RH-RS REGISTERS
1677
1678                    000250                  .=      250
1679    000250  000137  021476                  JMP     @#SRSWC         ;WORD COUNT REG
1680    000254  000137  021514                  JMP     @#SRSBA         ;BUS ADDR REG
1681    000260  000137  021532                  JMP     @#SRSDA         ;DISK ADDR REG
1682    000264  000137  021550                  JMP     @#SRSDS         ;DRIVE STATUS REG
1683    000270  000137  021566                  JMP     @#SRSER         ;ERROR REG
1684    000274  000137  021604                  JMP     @#SRSLA         ;LOOK AHEAD REG
1685    000300  000137  021614                  JMP     @#SRCS2         ;CS2 REG
1686    000304  000137  021632                  JMP     @#SRAS          ;ATTENTION SUMMARY REG
1687    000310  000137  021650                  JMP     @#RSMRR         ;MAINTENANCE REG
1688    000314  000137  021666                  JMP     @#SRSCS1        ;CS1 REG
1689
1690                    000320                  .=      320
1691    000320  052767  004000  000636          BIS     #BIT11,ONCEE    ;SET TKSEL TEST FLAG
1692    000326  000137  001206                  JMP     @#BEGIN         ;ENABLES OPERATOR TO DO A
1693                                                                    ;WRITE OR READ TO A DESIRED ADDR
1694                                                                    ;OPTIONAL WC, DA, AND UNIT #
1695
```

```
1696
1697  000001                    N=       1          ;INITALIZE FOR NEWTST
1698  104000                    HLT=     EMT        ;SET HLT TO EMT FOR ERROR TYPEOUTS
1699  177776                    PS=      177776     ;PROCESSOR STATUS
1700  177776                    PSW=     PS         ;PROCESSOR STATUS WORD
1701  177570                    SWR=     177570     ;SWITCH REGISTER
1702  177570                    DISPLAY=SWR         ;DISPLAY REGISTER
1703  000007                    BELL=    7          ;BELL
1704  000000                    R0=      %0         ;R0 - DEFINE REGISTERS
1705  000001                    R1=      %1         ;R1
1706  000002                    R2=      %2         ;R2
1707  000003                    R3=      %3         ;R3
1708  000004                    R4=      %4         ;R4
1709  000005                    R5=      %5         ;R5
1710  000006                    SP=      %6         ;R6 - STACK POINTER
1711  000007                    PC=      %7         ;R7 - PROGRAM COUNTER
1712  000001                    BIT0=    1          ;BIT EQUATES
1713  000002                    BIT1=    2
1714  000004                    BIT2=    4
1715  000010                    BIT3=    10
1716  000020                    BIT4=    20
1717  000040                    BIT5=    40
1718  000100                    BIT6=    100
1719  000200                    BIT7=    200
1720  000400                    BIT8=    400
1721  001000                    BIT9=    1000
1722  002000                    BIT10=   2000
1723  004000                    BIT11=   4000
1724  010000                    BIT12=   10000
1725  020000                    BIT13=   20000
1726  040000                    BIT14=   40000
1727  100000                    BIT15=   100000
1728  000001              GOOD=  %1                  ;FOR GOOD DATA
1729  000000              BAD=   %0                  ;FOR BAD DATA
1730
```

```
1731            001000                  .=      1000
1732
1733    001000  000000                  ICNT:   0                       ;LH = ITERATION COUNT  ;RH = TEST NO.
1734    001002  000000                  ERRORS: 0                       ;ERROR COUNT
1735    001004  000000  000000          PCNT:   0,0                     ;2 WORD PASS COUNT
1736    001010  000000                  LAD:    0                       ;LOOP ADDRESS FOR SCOPE
1737    001012  000000                  HLTADR: 0                       ;ADDRESS OF LAST HLT INSTRUCTION  EXECUTED
1738    001014  001000                  FILCHR: 1000                    ;FILCHR=0 (CHAR)  ;FILCHR+1=2 (COUNT)
1739    001016  177564                  TPS:    177564                  ;OUTPUT STATUS REGISTER
1740    001020  177566                  TPB:    177566                  ;OUTPUT BUFFER
1741
1742            001100                  .=      1100
1743    001100  000000                  SAVBAD: 0                       ;LOC FOR ILLEGAL FUNCTION CODE
1744    001102  000000                  OBUFSV: 0                       ;LOC OF OUTBUF
1745
1746                                    ;DISK I/O REGISTERS
1747
1748    001104  172040                  RSCS1:  172040                  ;DISK CONTROL + STATUS REGISTER
1749    001106  172050                  RSCS2:  172050                  ;DISK CONTROL + STATUS REGISTER
1750    001110  172042                  RSWC:   172042                  ;WORD COUNT REGISTER
1751    001112  172044                  RSBA:   172044                  ;BUS ADDRESS
1752    001114  172046                  RSDA:   172046                  ;DISK ADDRESS (DESIRED ADDRESS)
1753    001116  172052                  RSDS:   172052                  ;DRIVE STATUS
1754    001120  172054                  RSER:   172054                  ;ERROR REG.
1755    001122  172056                  RSAS:   172056                  ;ATTENTION SUMMARY
1756    001124  172060                  RSLA:   172060                  ;LOOK AHEAD
1757    001126  172062                  RSDB:   172062                  ;DATA BUFFER REGISTER
1758    001130  172064                  RSMR:   172064                  ;MAINTENANCE REGISTER
1759    001132  172066                  RSDT:   172066                  ;DRIVE TYPE REGISTER
1760    001134  172070                  RSBAE:  172070
1761    001136  172072                  RSCS3:  172072
1762    001140  000204                  RSVEC:  204                     ;INTERUPT VECTOR
1763    001142  000206                  RSVCPS: 206                     ;INTERRUPT PRIO. VECTOR
1764    001144  172041                  RSCS1B: 172041                  ;ODD BYTE ADD FOR CS1
1765    001146  172051                  RSCS2B: 172051                  ;ODD BYTE ADD FOR CS2
1766    001150  172043                  RSWCB:  172043                  ;ODD BYTE ADD FOR CW
1767    001152  172045                  RSBAB:  172045                  ;ODD BYTE ADD FOR BA
1768
1769                                    ;MEMORY MANAGEMENT REGISTER ASSIGNMENTS
1770
1771            177572                          SR0=177572
1772            172340                          KIPAR0=172340
1773            172342                          KIPAR1=172342
1774            172344                          KIPAR2=172344
1775            172356                          KIPAR7=172356
1776            172300                          KIPDR0=172300
1777            172302                          KIPDR1=172302
1778            172304                          KIPDR2=172304
1779            172316                          KIPDR7=172316
1780            000006                          RW=6
1781            000000                          UP=00
1782
```

```
                        ;BIT ASSIGNMENTS FOR ERROR TYPEOUTS
                        ;THE RS REGISTERS ARE DIVIDED INTO 3 GROUPS.
                        ;CS1,CS2 AND ER ARE IN THE FIRST GROUP.THIS GROUP IS ALWAYS
                        ;TYPED WITH EITHER OF THE OTHER GROUPS. AS,BA,DA, WC AND DS
                        ;ARE IN THE SECOND GROUP. DT,DB,MR, AND LA ARE IN THE 3RD
                        ;GROUP.YOU CAN NOT INTERMIX GROUP 2 OR 3. THEY HAVE
                        ;TO BE TYPED SEPERATELY.
                        ;EXAMPLE:   HLT !CS1,AS,BA
                        ;           HLT !CS1!DT!DB
                        ;
000001                      CS1=1           ;CONTROL AND STATUS 1
000002                      ER=2            ;CONTROL AND STATUS 2
000004                      DA=4            ;DESIRED ADD
000010                      WC=10           ;WORD COUNT
000020                      BA=20           ;BUS ADDRESS
000040                      DS=40           ;DRIVE STATUS
000100                      AS=100          ;ATTENTION SUMMARY
000200                      CS2=200         ;CONTROL AND STATUS REG
000204                      LA=204          ;LOOK AHEAD
000210                      DB=210          ;DATA BUFFER
000220                      MR=220          ;MAINTENANCE
000240                      DT=240          ;DRIVE TYPE

                        ;BIT ASSIGNMENTS FOR THE REGISTER BITS

040000                      TRE=40000       ;TRANSFER ERROR CS1
100000                      SC=100000       ;SPECIAL CONDITIONS CS1
000100                      IR=100          ;INPUT READY CS2
000200                      OR=200          ;OUTPUT READY CS2
002000                      PGE=2000        ;PROGRAM ERROR-CS2
010000                      NED=10000       ;NON-EXISTENT DRIVE CS2
040000                      WCE=40000       ;WRITE CHECK ERROR-CS2
100000                      DLT=100000      ;DATA LATE ERROR CS2
000200                      DRY=200         ;DRIVE READY DS
020000                      PIP=20000       ;POSITIONING IN PROGRESS DS
002000                      LBT=2000        ;LAST BLOCK TRANSFER-DS
040000                      ERR=40000       ;ERROR DS
100000                      ATA=100000      ;ATTENTION ACTIVE-DS
001000                      DAO=1000        ;DISK OVERFLOW ERROR-ER
100000                      DCK=100000      ;DATA CHECK ERROR-ER
000010                      BAI=10          ;BUS ADDR INCREMENT INHIBIT
000100                      IE=100          ;INTERRUPT INABLE CS1
```

```
                                ;WORKING LOCATIONS
                                
001154  146723                  RANNU:  146723                  ;RANDOM NUMBER PRIME
001156  000000                  UNNUM:  0                       ;UNIT CURRENTLY BEING TESTED
001160  000000                  UNITSV: 0                       ;SET BIT=UNIT ON BUS
001162  000000                  UNCMP:  0                       ;FOR COMPARING FOR # OF DEVICE
001164  000000                  ONCEE:  0                       ;DID WE TEST ANY DRIVES
001166  000000                  RS04DT: 0                       ;CLR IF RS03 SET IF RS04
001170  000000                  TIMSV:  0                       ;SAVE LOC FOR TIME
001172  000000                  ADBI:   0                       ;PORT SWITCH
        000004                  WWP=4                           ;WRITE WRONG PARITY
        172100                  MPRD=172100                     ;PARITY REG
001174  000000                  BPORTT: 0                       ;BUFFER ADDR FOR -B- PORT
001176  000000                  SAVEE:  0                       ;WORK LOC

                                ;DISCRIPTION OF ONCEE BITS
                                ;BIT0 MEANS FOUND DRIVE
                                ;BIT11 DO TKSEL TEST
                                ;BIT12 TYPE COULD NOT FIND NED ONLY ONCE
                                ;BIT13 TYPE NO MEM ON B PORT ONLY ONCE
                                ;BIT14 0- DO WCE WITH 0 -1 DO WCE WITH 1
                                ;BIT15 MEANS ERROR FOUND

                                ;RH11 WORK REGISTERS
                                ;(CAN BE CHANGED IN ANY ROUTINE)
001200  000000                  WORK:   0
001202  000000                  WORK1:  0
001204  000000                  WORK2:  0

001206  012706  000500  BEGIN:  MOV     #500,SP                 ;SET STACK TO  *** 500 ***
001212  012737  021040  000024          MOV     #.POWER,@#24            ;SET UP PF VECTOR
001220  012737  000340  000026          MOV     #340,@#26               ;LOCK OUT THE WORLD
001226  012737  020470  000030          MOV     #.HLT,@#30              ;SET EMT VECTOR
001234  012737  000340  000032          MOV     #340,@#32               ;LOCK UP
001242  012737  021440  000034          MOV     #.TRAP,@#34             ;SET TRAP VECTOR
001250  012737  000340  000036          MOV     #340,@#36               ;LOCK UP
001256  005067  177516                  CLR     ICNT                    ;INIT ICNT
001262  005067  177522                  CLR     LAD                     ;INIT LAD
001266  042767  143777  177670          BIC     #143777,ONCEE           ;CLEAR ONCEE
001274  012767  024550  177600          MOV     #OUTBUF,OBUFSV          ;SAVE LOC OF OUTBUFFER
001302  032767  004000  177654          BIT     #BIT11,ONCEE            ;DO TKSEL TEST?
001310  001402                          BEQ     .+6                     ;NO
001312  000137  021736                  JMP     @#TKSEL                 ;YES
```

```
                                        ;NOW TEST FOR DRIVES
                                        ;*************************************************************************
                                        ;TEST 1          TEST FOR DRIVES
                                        ;*************************************************************************
001316  104400                  TST1:   SCOPE

001320  012701  000010          MULTII: MOV     #8.,R1          ;PUT 8 INTO R1 FOR COUNT
001324  005077  177556                  CLR     @RSCS2          ;SET DEVICE TO ZERO
001330  012777  000007  177562  TRY:    MOV     #7.,@RSER       ;CAUSE AN ERROR +SETS BIT IN RSAS REG
001336  005301                          DEC     R1              ;DO A MAXIMUM OF 8 TIMES
001340  001403                          BEQ     DVNUM           ;TESTED FOR ALL DRIVES GET OUT
001342  005277  177540                  INC     @RSCS2          ;INCREMENT DRIVE UNIT
001346  000770                          BR      TRY             ;REPEAT FOR NEXT DRIVE
001350  017767  177546  177602  DVNUM:  MOV     @RSAS,UNITSV    ;SAVE
001356  012767  000401  177576          MOV     #401,UNCMP      ;SETUP TO CMP WITH UNITSV
001364  012767  000000  177564          MOV     #0,UNNUM        ;PUT 0 INTO UNIT NO.
001372  032767  020000  176170          BIT     #BIT13,SWR      ;INHIBIT TYPE OUT?
001400  001015                          BNE     STTEST          ;YES
001402  104402  001406                  TYPE    .+2             ;.ASCIZ <15><12>"TESTING UNIT "
001426  042767  100000  177530          BIC     #BIT15,ONCEE    ;CLEAR ERROR FLAG
001434  036767  177522  177516  STTEST: BIT     UNCMP,UNITSV    ;IS THIS DRIVE ON THE SYSTEM
001442  001461                          BEQ     TRYNX           ;NO
001444  016777  177506  177434          MOV     UNNUM,@RSCS2    ;YES PUT UNIT # INTO CS2
001452  022777  000000  177452          CMP     #0,@RSDT        ;IS THIS A RS03?
001460  001003                          BNE     2$              ;NO
001462  005067  177500                  CLR     RS04DT          ;YES
001466  000422                          BR      1$              ;GET OUT
001470  022777  000001  177434  2$:     CMP     #1,@RSDT        ;IS THIS A RS03 4US?
001476  001003                          BNE     3$              ;NO
001500  005067  177462                  CLR     RS04DT          ;RS03
001504  000413                          BR      1$              ;GET OUT
001506  022777  000002  177416  3$:     CMP     #2,@RSDT        ;IS THIS A RS04?
001514  001404                          BEQ     6$              ;YES
001516  022777  000003  177406          CMP     #3,@RSDT        ;IS IT A RS04?
001524  001030                          BNE     TRYNX           ;GET A NEW NUMBER
001526  052767  177777  177432  6$:     BIS     #-1,RS04DT      ;YES RS04
001534  032767  020000  176026  1$:     BIT     #BIT13,SWR      ;INHIBIT TYPE OUT?
001542  001020                          BNE     4$              ;YES
001544  032767  100000  177412          BIT     #BIT15,ONCEE    ;ANY ERRORS?
001552  001404                          BEQ     5$              ;NO
001554  104402  001560                  TYPE    ,.+2            ;.ASCIZ <15><12><12>
001564                          5$:
001564  016746  177366                  MOV     UNNUM,-(6)      ;PUT UNNUM ON STACK
001570  104406                          TYPES                   ;TYPE STACK IN OCTAL - SUPRESS
001572  104402  000040                  TYPE    ,40             ;TYPE SPACE
001576  042767  100000  177360          BIC     #BIT15,ONCEE    ;CLEAR ERROR FLAG
001604  000473                  4$:     BR      NOWGO           ;NOW TEST
```

```
1915  001606  006367  177350          TRYNX:  ASL     UNCMP           ;CHECK NEXT BIT FOR DRIVE
1916  001612  103403                          BCS     CHCKDV          ;DID WE TEST ANY REG?
1917  001614  005267  177336                  INC     UNNUM           ;INC UNIT #
1918  001620  000705                          BR      STTEST          ;CHECK FOR NEXT DRIVE
1919
1920  001622  032767  000001  177334  CHCKDV: BIT     #BIT0,ONCEE     ;DID WE TEST ANY DRIVES?
1921  001630  001057                          BNE     DONEE           ;YES WE DID TEST A DRIVE
1922  001632  012767  100000  177322          MOV     #100000,UNCMP   ;NO DRIVES TESTED, COULD NOT SET
1923  001640  005067  177312                  CLR     UNNUM           ;ANY AS BITS, THUS DEFAULTS TO
1924  001644  032767  020000  175716          BIT     #BIT13,SWR      ;INHIBIT TYPE OUT?
1925  001652  001045                          BNE     4$              ;YES
1926  001654  016746  177276                  MOV     UNNUM,-(6)      ;PUT UNNUM ON STACK
1927  001660  104406                          TYPES                   ;TYPE STACK IN OCTAL - SUPRESS
1928  001662  104402  000040                  TYPE    ,40             ;TYPE SPACE
1929  001666  104402  001672                  TYPE    ,.+2            ;.ASCIZ <15><12>"COULD NOT FIND DRIVE WILL TEST DRIVE 0
1930  001764  000000                          HALT                    ;WAIT
1931  001766  000402                  4$:     BR      NOWGO           ;TEST DRIVE 0
1932  001770  000167  016042          DONEE:  JMP     DONE            ;GET OUT
1933
1934                                  ;THIS TEST IS DESIGNED TO TEST THE ABILITY OF RESET
1935                                  ;TO CLEAR ALL THE RH AND RS REGISTERS
1936
1937  001774  012767  024550  177100  NOWGO:  MOV     #OUTBUF,OBUFSV  ;SAVE LOC OF OUTBUFFER
1938  002002  052767  000001  177154          BIS     #BIT0,ONCEE     ;SET FOUND DRIVE FLAG
1939  002010  016767  016452  177152          MOV     TIMES,TIMSV     ;SAVE TIME
1940  002016  012767  000001  016442          MOV     #1,TIMES        ;ONLY TEST ONCE
1941                                  ;*****************************************************************************
1942                                  ;TEST 2         RESET TEST FOR REGISTERS
1943                                  ;*****************************************************************************
1944  002024  104400                  TST2:   SCOPE
1945  002026  012737  000340  177776          MOV     #340,@#PS       ;LOCK OUT INTERUPTS
1946  002034  016777  177116  177044          MOV     UNNUM,@RSCS2    ;GET UNIT #
1947  002042  012777  177776  177034          MOV     #177776,@RSCS1  ;SET ALL
1948  002050  012777  177777  177034          MOV     #177777,@RSBA   ;POSSIBLE R/W
1949  002056  012777  177777  177030          MOV     #177777,@RSDA   ;BITS IN THESE REGISTERS
1950  002064  012777  177777  177026          MOV     #177777,@RSER
1951  002072  012777  177777  177030          MOV     #177777,@RSMR
1952  002100  012777  177777  177002          MOV     #177777,@RSWC
1953  002106  012777  177737  176772          MOV     #177737,@RSCS2
1954  002114  000005                          RESET                   ;CLEAR ALL BITS IN ALL REG.
1955
1956                                  ;TEST RSCS2 FOR CLEARED BITS
1957
1958  002116  022777  000100  176762          CMP     #100,@RSCS2     ;DID THESE BITS GET CLEARED?
1959  002124  001401                          BEQ     .+4             ;YES
1960  002126  104200                          HLT     !CS2            ;(417) SHOULD BE CLEARED IN CS2
1961  002130  016777  177022  176750          MOV     UNNUM,@RSCS2    ;PUT # OF UNIT IN TEST IN CS2
1962  002136  022777  010600  176752          CMP     #10600,@RSDS    ;IS DPR AND MOL SET?
1963  002144  001401                          BEQ     .+4             ;YES
1964  002146  104040                          HLT     !DS             ;NO WHY NOT?
1965                                  ;TEST CONTROL AND STATUS REG 1
1966  002150  022777  004200  176726          CMP     #4200,@RSCS1    ;DID THE READY BIT SET?
1967  002156  001401                          BEQ     .+4             ;YES
1968  002160  104001                          HLT     !CS1            ;READY SHOULD BE SET
```

```
                                        ;TEST BUS ADDRESS REGISTER

002162  005777  176724                        TST     @RSBA           ;IS BA REG. CLEARED
002166  001401                                BEQ     .+4             ;YES
002170  104020                                HLT     !BA             ;SHOULD BE 0

                                        ;TEST DISK ADDRESS REGISTER

002172  005777  176716                        TST     @RSDA           ;IS DA CLEARED
002176  001401                                BEQ     .+4             ;YES
002200  104004                                HLT     !DA             ;SHOULD BE 0

                                        ;TEST ERROR REG RSER

002202  005777  176712                        TST     @RSER           ;DID RSER CLEAR?
002206  001401                                BEQ     .+4             ;YES
002210  104002                                HLT     !ER             ;BITS(157015) SHOULD BE CLEARED

                                        ;TEST RS MAINTENANCE REGISTER

002212  032777  000077  176710                BIT     #77,@RSMR       ;DID THESE BITS GET CLEARED
002220  001401                                BEQ     .+4             ;YES
002222  104220                                HLT     !MR             ;BITS(77) SHOULD BE 0

                                        ;TEST WC REG IT SHOULD NOT CHANGE

002224  022777  177777  176656                CMP     #177777,@RSWC   ;DID IT CHANGE?
002232  001401                                BEQ     .+4             ;NO
002234  104010                                HLT     !WC             ;RESET SHOULD NOT MODIFY RSWC

                                        ;TEST RSAS

002236  005777  176660                        TST     @RSAS           ;IS REG CLEAR
002242  001401                                BEQ     .+4             ;YES
002244  104100                                HLT     !AS             ;NO
```

```
                                ;*****************************************************************************
                                ;TEST 3          TEST CLEAR BIT IN CS2 ON ALL THE R/W BITS
                                ;*****************************************************************************
002246  104400                  TST3:   SCOPE

002250  012737  000340  177776  TTAGG:  MOV     #340,@#PS       ;LOCK OUT INTERRUPTS
002256  016777  176674  176622          MOV     UNNUM,@RSCS2    ;GET UNIT #
002264  012777  043576  176612          MOV     #43576,@RSCS1   ;SET ALL
002272  012777  177777  176612          MOV     #177777,@RSBA   ;POSSIBLE
002300  012777  177777  176606          MOV     #177777,@RSDA   ;REGISTERS
002306  012777  177777  176604          MOV     #177777,@RSER
002314  012777  177777  176604          MOV     #177777,@RSDB
002322  012777  177777  176560          MOV     #177777,@RSWC
002330  012777  020417  176550          MOV     #20417,@RSCS2
002336  012777  000071  176564          MOV     #71,@RSMR
002344  012777  000040  176534          MOV     #40,@RSCS2      ;CLEAR ALL BITS
002352  022777  000100  176526          CMP     #100,@RSCS2     ;DID THE RIGHT BITS CLEAR?
002360  001401                          BEQ     .+4             ;YES
002362  104200                          HLT     !CS2            ;(417) SHOULD BE CLEARED IN CS2
002364  016777  176566  176514          MOV     UNNUM,@RSCS2    ;GET DRIVE NUMBER
002372  032777  173577  176504          BIT     #173577,@RSCS1  ;DID ALL BITS GET CLEARED
002400  001401                          BEQ     .+4             ;YES
002402  104001                          HLT     !CS1            ;NO, ALL BITS SHOULD BE 0
                                ;TEST BUS ADDRESS REGISTER

002404  005777  176502                  TST     @RSBA           ;IS BA REG. CLEARED
002410  001401                          BEQ     .+4             ;YES
002412  104020                          HLT     !BA             ;SHOULD BE 0

                                ;TEST DISK ADDRESS REGISTER

002414  005777  176474                  TST     @RSDA           ;IS DA CLEARED
002420  001401                          BEQ     .+4             ;YES
002422  104020                          HLT     !BA             ;SHOULD BE 0

                                ;TEST ERROR REG RSER

002424  005777  176470                  TST     @RSER           ;DID THESE BITS GET CLEARED
002430  001401                          BEQ     .+4             ;YES
002432  104002                          HLT     !ER             ;BITS(157015) SHOULD BE CLEARED

                                ;TEST RS MAINTENANCE REGISTER
002434  032777  000077  176466          BIT     #77,@RSMR       ;DID THESE BITS GET CLEARED
002442  001401                          BEQ     .+4             ;YES
002444  104220                          HLT     !MR             ;BITS(77) SHOULD BE 0

                                ;TEST WC REG. IT SHOULD NOT CHANGE
002446  022777  177777  176434          CMP     #177777,@RSWC   ;DID WC CHANGE
002454  001401                          BEQ     .+4             ;NO
002456  104010                          HLT     !WC             ;WHY DID IT CHANGE?
```

```
2054                                    ;*****************************************************************************
2055                                    ;TEST 4          SET AND CLEAR ALL REGISTERS
2056                                    ;*****************************************************************************
2057  002460  104400                    TST4:   SCOPE
2058                                    ;CAN WE SET THE FUNCTION BITS IN THE RSCS1 REG.
2059                                    ;BITS 7,6,5,4,3,2&1
2060
2061  002462  104414                            CLRDK                   ;CLEAR ALL RS REG
2062  002464  016767  176500  015774            MOV     TIMSV,TIMES     ;GET TIME
2063  002472  012777  003576  176404            MOV     #3576,@RSCS1    ;SET DISK FUNCTION BITS
2064  002500  022777  005776  176376            CMP     #5776,@RSCS1    ;ARE THESE BITS SET?
2065  002506  001401                            BEQ     .+4             ;NO
2066  002510  104001                            HLT     !CS1            ;SHOULD = 3776
2067  002512  012777  002524  176364            MOV     #2524,@RSCS1    ;SET THESE BITS
2068  002520  022777  004724  176356            CMP     #4724,@RSCS1    ;DID THEY SET
2069  002526  001401                            BEQ     .+4             ;YES
2070  002530  104001                            HLT     !CS1            ;SHOULD BE 2725
2071  002532  012777  001052  176344            MOV     #1052,@RSCS1    ;SET THESE BITS
2072  002540  022777  005252  176336            CMP     #5252,@RSCS1    ;ARE THEY =?
2073  002546  001401                            BEQ     .+4             ;YES
2074  002550  104001                            HLT     !CS1            ;SHOULD = 1252
2075  002552  104400                    TST5:   SCOPE
2076                                    ;CLEAR THE FUNCTION BITS
2077
2078  002554  012777  043576  176322            MOV     #43576,@RSCS1   ;SET DISK FUNCTION BITS
2079  002562  005077  176316                    CLR     @RSCS1
2080  002566  022777  004200  176310            CMP     #4200,@RSCS1    ;IS THE READY BIT SET
2081  002574  001401                            BEQ     .+4             ;YES
2082  002576  104001                            HLT     !CS1            ;RSCS1 SHOULD = 4200
2083
2084                                    ;*****************************************************************************
2085                                    ;TEST 6          TEST RSCS2
2086                                    ;*****************************************************************************
2087  002600  104400                    TST6:   SCOPE
2088
2089  002602  000005                            RESET                   ;CLEAR WORLD
2090  002604  022777  000100  176274            CMP     #100,@RSCS2     ;DID THEY CLEAR?
2091  002612  001401                            BEQ     .+4             ;YES
2092  002614  104200                            HLT     !CS2            ;NO
2093  002616  012777  021037  176262            MOV     #21037,@RSCS2   ;SET BITS 21017
2094  002624  022777  000137  176254            CMP     #137,@RSCS2     ;DID THESE BITS GET SET
2095  002632  001405                            BEQ     1$              ;YES
2096  002634  017700  176246                    MOV     @RSCS2,BAD
2097  002640  012701  000137                    MOV     #137,GOOD       ;WHAT CS2 SHOULD =
2098  002644  104000                            HLT                     ;CS2 = BAD GOOD = CORRECT ANS
```

```
2099  002646  012777  020025  176232  1$:     MOV     #20025,@RSCS2   ;SET THESE BITS
2100  002654  022777  000125  176224          CMP     #125,@RSCS2     ;DID THESE BITS GET SET
2101  002662  001401                          BEQ     .+4             ;YES
2102  002664  104200                          HLT     !CS2            ;NO,CS2 SHOULD = 20125
2103  002666  012777  000012  176212          MOV     #12,@RSCS2      ;LOAD THESE BITS
2104  002674  022777  000112  176204          CMP     #112,@RSCS2     ;DID THESE BITS GET SET IN CS2
2105  002702  001401                          BEQ     .+4             ;YES
2106  002704  104200                          HLT     !CS2            ;BAD = CS2 GOOD = CORRECT ANS
2107  002706  012777  177777  176172          MOV     #-1,@RSCS2      ;SET BITS
2108  002714  005077  176166                  CLR     @RSCS2          ;CLEAR THEM
2109  002720  022777  000100  176160          CMP     #100,@RSCS2     ;DID CLEAR WORK
2110  002726  001401                          BEQ     .+4             ;YES
2111  002730  104200                          HLT     !CS2            ;R/W BITS DID NOT CLEAR
2112  002732  016777  176220  176146          MOV     UNNUM,@RSCS2    ;GET UNIT #
2113  002740  104400                  TST7:   SCOPE
2114                                  ;CAN WE SET ALL THE RSBA BITS
2115
2116  002742  012777  177777  176142          MOV     #177777,@RSBA   ;SET THE BITS
2117  002750  022777  177776  176134          CMP     #177776,@RSBA   ;DID THEY SET
2118  002756  001401                          BEQ     .+4             ;YES
2119  002760  104020                          HLT     !BA             ;BITS 17776 SHOULD BE SET
2120  002762  012777  125252  176122          MOV     #125252,@RSBA   ;SET THESE BITS
2121  002770  022777  125252  176114          CMP     #125252,@RSBA   ;ARE THEY =
2122  002776  001401                          BEQ     .+4             ;YES
2123  003000  104020                          HLT     !BA             ;SHOULD BE 125252
2124  003002  012777  052524  176102          MOV     #52524,@RSBA    ;SET THESE BITS
2125  003010  022777  052524  176074          CMP     #52524,@RSBA    ;ARE THEY =
2126  003016  001401                          BEQ     .+4             ;YES
2127  003020  104020                          HLT     !BA             ;SHOULD BE 52524
2128
2129  003022  104400                  TST10:  SCOPE
2130                                  ;FLOAT A 1 THROUGH RSBA
2131
2132  003024  012701  000002          FLOTBA: MOV     #2,GOOD         ;GET A 2
2133  003030  000241                          CLC                     ;CLEAR CARRY
2134  003032  010177  176054          1$:     MOV     GOOD,@RSBA      ;FLOAT NUMBER
2135  003036  017700  176050                  MOV     @RSBA,BAD       ;GET BA
2136  003042  020100                          CMP     GOOD,BAD        ;COMPARE BA
2137  003044  001401                          BEQ     .+4             ;BA CORRECT
2138  003046  104000                          HLT                     ;BAD=BA GOOD=CORRECT AN
2139  003050  006101                          ROL     GOOD            ;ROTATE NUMBER
2140  003052  103367                          BCC     1$              ;LOOP TILL DONE
```

```
2141  003054  104400                  TST11:  SCOPE
2142
2143                                  ;CLEAR THE RSBA REGISTER
2144
2145  003056  012777  177777  176026          MOV     #177777,@RSBA   ;SET RSBA EQUAL TO ALL ONES
2146  003064  005077  176022                  CLR     @RSBA
2147  003070  005777  176016                  TST     @RSBA           ;TEST FOR BIT0 SET IN RSBA (READ ONLY BIT)
2148  003074  001401                          BEQ     .+4             ;YES
2149  003076  104020                          HLT     !BA             ;NO
2150  003100  104400                  TST12:  SCOPE
2151
2152                                  ;CAN WE SET ALL BITS IN RSWC REGISTER
2153
2154  003102  012777  177777  176000          MOV     #177777,@RSWC   ;SET WC BITS
2155  003110  022777  177777  175772          CMP     #177777,@RSWC   ;ARE ALL BITS SET
2156  003116  001401                          BEQ     .+4             ;YES
2157  003120  104010                          HLT     !WC             ;NO
2158  003122  012777  125252  175760          MOV     #125252,@RSWC   ;SET THESE BITS
2159  003130  022777  125252  175752          CMP     #125252,@RSWC   ;ARE THEY =
2160  003136  001401                          BEQ     .+4             ;YES
2161  003140  104010                          HLT     !WC             ;SHOULD BE 125252
2162  003142  012777  052525  175740          MOV     #52525,@RSWC    ;SET THESE BITS
2163  003150  022777  052525  175732          CMP     #52525,@RSWC    ;ARE THEY =
2164  003156  001401                          BEQ     .+4             ;YES
2165  003160  104010                          HLT     !WC             ;SHOULD BE 152525
2166  003162  104400                  TST13:  SCOPE
2167
2168                                  ;FLOAT A 1 THROUGH RSWC
2169
2170  003164  012701  000001          FLOTWC: MOV     #1,GOOD         ;GET A 1
2171  003170  000241                          CLC                     ;CLEAR CARRY
2172  003172  010177  175712          1$:     MOV     GOOD,@RSWC      ;FLOAT NUMBER
2173  003176  017700  175706                  MOV     @RSWC,BAD       ;GET WC
2174  003202  020100                          CMP     GOOD,BAD        ;COMPARE WC
2175  003204  001401                          BEQ     .+4             ;WC CORRECT
2176  003206  104000                          HLT                     ;BAD=WC GOOD=CORRECT ANS
2177  003210  006101                          ROL     GOOD            ;ROTATE NUMBER
2178  003212  103367                          BCC     1$              ;LOOP TILL DONE
```

```
                                        ;CLEAR THE WORD COUNT REGISTER
003214  104400                  TST14:  SCOPE

003216  012777  177777  175664          MOV     #177777,@RSWC   ;SET RSWC REGISTER EQUAL TO ALL ONES
003224  005077  175660                  CLR     @RSWC
003230  005777  175654                  TST     @RSWC           ;DID ALL BITS GET CLEARED
003234  001401                          BEQ     .+4             ;YES
003236  104010                          HLT     !WC             ;NO
003240  104400                  TST15:  SCOPE

                                        ;CAN WE SET ALL THE BITS IN THE RSDA REGISTER.

003242  012777  177777  175644          MOV     #177777,@RSDA   ;SET ALL BITS
003250  022777  177777  175636          CMP     #177777,@RSDA   ;ARE THE BITS SET
003256  001401                          BEQ     .+4             ;YES
003260  104004                          HLT     !DA             ;NO
003262  012777  125252  175624          MOV     #125252,@RSDA   ;SET THESE BITS
003270  022777  125252  175616          CMP     #125252,@RSDA   ;ARE THEY =
003276  001401                          BEQ     .+4             ;YES
003300  104004                          HLT     !DA             ;SHOULD BE 125252
003302  012777  052525  175604          MOV     #52525,@RSDA    ;SET THESE BITS
003310  022777  052525  175576          CMP     #52525,@RSDA    ;ARE THEY =
003316  001401                          BEQ     .+4             ;YES
003320  104004                          HLT     !DA             ;SHOULD BE 52525
003322  104400                  TST16:  SCOPE

                                        ;FLOAT A 1 THROUGH RSDA

003324  012701  000001          FLOTDA: MOV     #1,GOOD         ;GET A 1
003330  000241                          CLC                     ;CLEAR CARRY
003332  010177  175556          1$:     MOV     GOOD,@RSDA      ;FLOAT NUMBER
003336  017700  175552                  MOV     @RSDA,BAD       ;GET DA
003342  020100                          CMP     GOOD,BAD        ;COMPARE DA
003344  001401                          BEQ     .+4             ;DA CORRECT
003346  104000                          HLT                     ;BAD=DA GOOD=CORRECT ANS
003350  006101                          ROL     GOOD            ;ROTATE NUMBER
003352  103367                          BCC     1$              ;LOOP TILL DONE
```

```
2216                                        ;CAN WE CLEAR THE RSDA REG.
2217  003354  104400                  TST17:  SCOPE
2218
2219  003356  012777  177777  175530        MOV     #177777,@RSDA   ;SET RSDA TO ALL ONES
2220  003364  005077  175524                CLR     @RSDA
2221  003370  005777  175520                TST     @RSDA           ;TEST FOR ZERO RSDA
2222  003374  001401                        BEQ     .+4             ;YES
2223  003376  104004                        HLT     !DA             ;ANS SHOULD BE 0
2224  003400  104400                  TST20:  SCOPE
2225
2226                                        ;SET AND CLEAR THE RSER REG.
2227
2228  003402  012777  177777  175510        MOV     #177777,@RSER   ;SET THESE BITS
2229  003410  022777  177017  175502        CMP     #177017,@RSER   ;DID THEY SET
2230  003416  001401                        BEQ     .+4             ;YES
2231  003420  104002                        HLT     !ER             ;RSER SHOULD = 157017
2232  003422  112777  000001  175470        MOVB    #1,@RSER        ;A MOVB INST
2233  003430  022777  000001  175462        CMP     #1,@RSER        ;SHOULD MODIFY COMPLETE WD
2234  003436  001401                        BEQ     .+4             ;OK
2235  003440  104002                        HLT     !ER
2236
2237  003442  104400                  TST21:  SCOPE
2238
2239  003444  012777  052005  175446        MOV     #52005,@RSER    ;SET THESE BITS
2240  003452  022777  052005  175440        CMP     #52005,@RSER    ;DID THEY SET
2241  003460  001401                        BEQ     .+4             ;YES
2242  003462  104002                        HLT     !ER             ;ER SHOULD = 52005
2243  003464  104400                  TST22:  SCOPE
2244
2245  003466  012777  125012  175424        MOV     #125012,@RSER   ;SET THESE BITS
2246  003474  022777  125012  175416        CMP     #125012,@RSER   ;DID THEY SET
2247  003502  001401                        BEQ     .+4             ;YES
2248  003504  104002                        HLT     !ER             ;ER SHOULD = 105012
```

```
2249  003506  104400                    TST23:  SCOPE
2250
2251  003510  012777  177017  175402            MOV     #177017,@RSER   ;SET THESE BITS
2252  003516  005077  175376                    CLR     @RSER           ;CLEAR THEM
2253  003522  005777  175372                    TST     @RSER           ;DID THEY CLEAR
2254  003526  001401                            BEQ     .+4             ;YES
2255  003530  104002                            HLT     !ER             ;SHOULD = 0
2256  003532  104400                    TST24:  SCOPE
2257
2258                                    ;SET AND CLEAR RSMR
2259
2260  003534  012777  000070  175366            MOV     #70,@RSMR       ;SET THESE BITS
2261  003542  017767  175362  175430            MOV     @RSMR,WORK      ;PUT INTO WORKABLE REG
2262  003550  042767  177700  175422            BIC     #177700,WORK    ;CLEAR JUNK
2263  003556  022767  000070  175414            CMP     #70,WORK        ;DID THEY SET
2264  003564  001401                            BEQ     .+4             ;YES
2265  003566  104220                            HLT     !MR             ;SHOULD = 70
2266  003570  104400                    TST25:  SCOPE
2267
2268  003572  012777  000070  175330            MOV     #70,@RSMR       ;SET BITS
2269  003600  005077  175324                    CLR     @RSMR           ;CLEAR THEM
2270  003604  032777  000077  175316            BIT     #77,@RSMR       ;DID THEY CLEAR
2271  003612  001401                            BEQ     .+4             ;YES
2272  003614  104220                            HLT     !MR             ;BITS (77) SHOULD = 0
2273  003616  104400                    TST26:  SCOPE
2274
2275  003620  012777  000050  175302            MOV     #50,@RSMR       ;SET BITS
2276  003626  017767  175276  175344            MOV     @RSMR,WORK      ;PUT IN WORKABLE REG
2277  003634  042767  177700  175336            BIC     #177700,WORK    ;CLEAR JUNK
2278  003642  022767  000050  175330            CMP     #50,WORK        ;DID THESE BITS SET
2279  003650  001401                            BEQ     .+4             ;YES
2280  003652  104220                            HLT     !MR             ;BITS (50) SHOULD BE SET
2281  003654  104400                    TST27:  SCOPE
2282
2283  003656  012777  000020  175244            MOV     #20,@RSMR       ;SET BITS
2284  003664  017767  175240  175306            MOV     @RSMR,WORK      ;PUT INTO WORKABLE REG
2285  003672  042767  177700  175300            BIC     #177700,WORK    ;CLEAR JUNK
2286  003700  022767  000020  175272            CMP     #20,WORK        ;DID THEY SET
2287  003706  001401                            BEQ     .+4             ;YES
2288  003710  104220                            HLT     !MR             ;MR SHOULD AT LEAST HAVE A (21)
```

```
                                ;*****************************************************************************
                                ;TEST 30                    LOAD RANDOM NUMBERS INTO RSWC, RSDA AND RSBA
                                ;*****************************************************************************
003712  104400                  TST30:  SCOPE
                                
003714  012767  001000  175256  RANTS:  MOV     #1000,WORK      ;MAKE TABLE 1000 WDS LONG
003722  012701  024550                  MOV     #OUTBUF,R1      ;GET SRARTING LOC OF TABLE
003726  004567  020360                  JSR     R5,RANDOM       ;GENERATE #
003732  012704  024550                  MOV     #OUTBUF,R4      ;SETUP FOR COMPARE
003736  012767  003744  175044          MOV     #LOP1,LAD       ;SETUP LOOP ADDR
003744  012703  001000          LOP1:   MOV     #1000,R3        ;LOAD TEST COUNTER
003750  005303                  4$:     DEC     R3              ;DONE WITH COMPARE?
003752  001413                          BEQ     1$              ;YES
003754  016705  175130                  MOV     RSWC,R5         ;GET WC ADDRESS
003760  011415                          MOV     (R4),(R5)       ;LOAD WC
003762  021524                          CMP     (R5),(R4)+      ;IS IT CORRECT?
003764  001771                          BEQ     4$              ;YES
003766  017700  175116                  MOV     @RSWC,BAD       ;GET BAD WC
003772  014401                          MOV     -(R4),GOOD      ;GET GOOD ANS
003774  104000                          HLT                     ;TYPE THEM OUT
003776  005724                          TST     (R4)+           ;UPDATE RANDOM NUMBER
004000  000763                          BR      4$              ;CONT
004002  012704  024550          1$:     MOV     #OUTBUF,R4      ;GET STARTING LOC OF TABLE
004006  012767  004014  174774          MOV     #LOP2,LAD       ;SETUP LOOP ADDR
004014  012703  001000          LOP2:   MOV     #1000,R3        ;SETUP TEST COUNTER
004020  005303                  3$:     DEC     R3              ;DONE YET?
004022  001413                          BEQ     1$              ;YES
004024  016705  175064                  MOV     RSDA,R5         ;LOAD DA ADDRESS INTO R5
004030  011415                          MOV     (R4),(R5)       ;LOAD DA
004032  021524                          CMP     (R5),(R4)+      ;IS IT CORRECT?
004034  001771                          BEQ     3$              ;YES
004036  017700  175052                  MOV     @RSDA,BAD       ;GET BAD DATA
004042  014401                          MOV     -(R4),GOOD      ;GET GOOD DATA
004044  104000                          HLT                     ;TYPE IT OUT
004046  005724                          TST     (R4)+           ;UPDATE RANDOM NUMBER
004050  000763                          BR      3$              ;CONTINUE
004052  012704  024550          1$:     MOV     #OUTBUF,R4      ;GET STARTING LOC OF TABLE
004056  012767  004064  174724          MOV     #LOP3,LAD       ;SETUP LOOP ADDR
004064  012703  001000          LOP3:   MOV     #1000,R3        ;SETUP TEST COUNTER
004070  005303                  3$:     DEC     R3              ;DONE YET?
004072  001416                          BEQ     2$              ;YES
004074  016705  175012                  MOV     RSBA,R5         ;LOAD ADDRESS OF BA INTO R5
004100  011415                          MOV     (R4),(R5)       ;LOAD BA
004102  042714  000001                  BIC     #BIT0,(R4)      ;CLEAR BIT 0
004106  021514                          CMP     (R5),(R4)       ;IS IT CORRECT?
004110  001767                          BEQ     3$              ;YES
004112  017700  174774                  MOV     @RSBA,BAD       ;GET BAD DATA
004116  011401                          MOV     (R4),GOOD       ;GET GOOD DATA
004120  104000                          HLT                     ;TYPE IT OUT
004122  000400                          BR      1$              ;GET OUT
004124  005724                  1$:     TST     (R4)+           ;GET NEW NUMBER
004126  000760                          BR      3$              ;CONTINUE
004130  000240                  2$:     NOP
```

```
                                ;*****************************************************************************
                                ;TEST 31                    TEST ODD BYTE INSTRUCTIONS ON CS1, CS2, WC AND BA
                                ;*****************************************************************************
004132  104400                  TST31:  SCOPE

004134  104414                  BITST:  CLRDK                           ;CLEAR ALL RS REG
004136  012777  003566  174740          MOV     #3566,@RSCS1            ;LOAD CS1
004144  112777  000005  174772          MOVB    #5,@RSCS1B              ;LOAD BIT
004152  022777  004766  174724          CMP     #4766,@RSCS1            ;DID IT LOAD?
004160  001401                          BEQ     .+4                     ;YES
004162  104001                          HLT     !CS1
004164  112777  000032  174712          MOVB    #32,@RSCS1
004172  022777  004632  174704          CMP     #4632,@RSCS1
004200  001401                          BEQ     .+4
004202  104001                          HLT     !CS1                    ;CS1 SHOULD = 4632

004204  104400                  TST32:  SCOPE

004206  016777  174744  174672  BITCS2: MOV     UNNUM,@RSCS2            ;LOAD UNIT NUMBER
004214  052777  177400  174664          BIS     #177400,@RSCS2          ;LOAD ALL BITS
004222  105077  174720                  CLRB    @RSCS2B                 ;CLR UPPER BYTE
004226  016701  174724                  MOV     UNNUM,GOOD              ;GET UNIT NO.
004232  052701  000100                  BIS     #100,GOOD               ;SET OR BIT
004236  017700  174644                  MOV     @RSCS2,BAD              ;GET CS2
004242  020001                          CMP     BAD,GOOD                ;IS CS2 CORRECT?
004244  001401                          BEQ     .+4                     ;YES
004246  104000                          HLT                             ;LOAD BYTE DID NOT WORK

004250  104400                  TST33:  SCOPE

004252  012777  025252  174630  BITWC:  MOV     #25252,@RSWC            ;LOAD WC
004260  112777  000377  174662          MOVB    #377,@RSWCB             ;LOAD BIT
004266  022777  177652  174614          CMP     #177652,@RSWC           ;DID IT LOAD?
004274  001401                          BEQ     .+4                     ;YES
004276  104010                          HLT     !WC                     ;NO WC SHOULD =177652
004300  112777  000123  174602          MOVB    #123,@RSWC
004306  022777  177523  174574          CMP     #177523,@RSWC
004314  001401                          BEQ     .+4
004316  104010                          HLT     !WC                     ;WC SHOULD = 177523

004320  104400                  TST34:  SCOPE

004322  012777  025252  174562  BITBA:  MOV     #25252,@RSBA            ;LOAB DA
004330  112777  000377  174614          MOVB    #377,@RSBAB             ;LOAD BIT
004336  022777  177652  174546          CMP     #177652,@RSBA           ;DID IT LOAD?
004344  001401                          BEQ     .+4                     ;YES
004346  104020                          HLT     !BA                     ;DA SHOULD =177652
004350  112777  000125  174534          MOVB    #125,@RSBA
004356  022777  177524  174526          CMP     #177524,@RSBA
004364  001401                          BEQ     .+4
004366  104020                          HLT     !BA                     ;BA SHOULD = 177525
004370  104414                          CLRDK                           ;CLEAR ALL RS REG
```

```
2394  004372  104400                TST35:  SCOPE
2395  004374  104414                ZERONE: CLRDK                           ;CLEAR ALL RS REG
2396  004376  005077  174524                CLR     @RSDB                   ;LOAD DB WITH ALL 0
2397  004402  012777  177777  174516        MOV     #177777,@RSDB           ;LOAD DB WITH ALL ONES
2398  004410  012767  002000  174562        MOV     #2000,WORK              ;TIME OUT ROUTINE
2399  004416  012701  000300                MOV     #300,GOOD               ;GET CORRECT FOR CS2
2400  004422  056701  174530                BIS     UNNUM,GOOD
2401  004426  017700  174454          2$:   MOV     @RSCS2,BAD              ;GET CS2
2402  004432  020100                      CMP     GOOD,BAD                ;IS IT CORRECT?
2403  004434  001404                      BEQ     3$                      ;YES
2404  004436  005367  174536              DEC     WORK                    ;TO WAIT FOR OR
2405  004442  001371                      BNE     2$                      ;TO SET
2406  004444  104200                      HLT     !CS2                    ;OR SHOULD BE SET
2407  004446  005001                3$:   CLR     GOOD
2408  004450  017700  174452              MOV     @RSDB,BAD               ;LOAD BAD WITH DB
2409  004454  020100                      CMP     GOOD,BAD                ;IS BAD CORRECT
2410  004456  001401                      BEQ     .+4                     ;YES
2411  004460  104000                      HLT                             ;COULD NOT FLOAT 0 THROUGH DB
2412  004462  012701  177777              MOV     #-1,GOOD                ;LOAD GOOD WITH ANS
2413  004466  017700  174434              MOV     @RSDB,BAD               ;GET DATA FROM DB
2414  004472  020100                      CMP     GOOD,BAD                ;IS DB CORRECT
2415  004474  001401                      BEQ     .+4                     ;YES
2416  004476  104000                      HLT                             ;BAD SHOULD = 1777777
```

```
2417                                    ;*****************************************************************************
2418                                    ;TEST 36                    TEST NO-OP FUNCTION
2419                                    ;*****************************************************************************
2420  004500  104400                    TST36:  SCOPE
2421
2422  004502  104414                    NOOP:   CLRDK                           ;CLEAR ALL RS REG
2423  004504  012767  177777  020036            MOV     #177777,OUTBUF          ;DATA TO BE XFERED
2424  004512  013777  001102  174372            MOV     @#OBUFSV,@RSBA          ;SET UP CURRENT ADDRESS
2425  004520  012777  177777  174362            MOV     #-1,@RSWC               ;LOAD WC WITH -1
2426  004526  012777  000001  174350            MOV     #1,@RSCS1               ;DO NO-OP FUNCTION
2427  004534  032777  000001  174342            BIT     #1,@RSCS1               ;DID GO BIT CLEAR
2428  004542  001401                            BEQ     .+4                     ;YES
2429  004544  104001                            HLT     !CS1                    ;GO BIT SHOULD BE CLEARED
2430  004546  005777  174346                    TST     @RSER                   ;DID ANY ERRORS OCCURR?
2431  004552  001401                            BEQ     .+4                     ;NO
2432  004554  104002                            HLT     !ER                     ;ALL ERROR BITS SHOULD BE 0
2433  004556  022777  177777  174324            CMP     #-1,@RSWC               ;DID WC MOVE?
2434  004564  001401                            BEQ     .+4                     ;NO
2435  004566  104010                            HLT     !WC                     ;WC SHOULD = 1777777
2436  004570  005777  174320                    TST     @RSDA                   ;DID DA MOV
2437  004574  001401                            BEQ     .+4                     ;NO
2438  004576  104004                            HLT     !DA                     ;DA SHOULD =0
2439  004600  023777  001102  174304            CMP     @#OBUFSV,@RSBA          ;DID BA MOVE
2440  004606  001401                            BEQ     .+4                     ;NO
2441  004610  104020                            HLT     !BA                     ;BA MOVED
2442  004612  036777  174344  174302            BIT     UNCMP,@RSAS             ;AS SHOULD NOT SET ON
2443  004620  001401                            BEQ     .+4                     ;A NO-OP FUNCTION
2444  004622  104100                            HLT     !AS                     ;AS SET WHY?
```

```
2445                                    ;*****************************************************************************
2446                                    ;TEST 37                    TEST NO-OP FUNCTION WITH ERROR BITS SET
2447                                    ;*****************************************************************************
2448  004624  104400                    TST37:  SCOPE
2449
2450  004626  104414                    NNOOP:  CLRDK                           ;CLEAR ALL RS REG
2451  004630  012777  000007  174262            MOV     #7,@RSER                ;LOAD ER
2452  004636  036777  174320  174256            BIT     UNCMP,@RSAS             ;IS ATA BIT SET?
2453  004644  001001                            BNE     .+4                     ;YES
2454  004646  104100                            HLT     !AS                     ;AS BIT SHOULD BE SET
2455  004650  012767  177777  017672            MOV     #177777,OUTBUF          ;DATA TO BE XFERED
2456  004656  013777  001102  174226            MOV     @#OBUFSV,@RSBA          ;SET UP CURRENT ADDRESS
2457  004664  012777  177777  174216            MOV     #-1,@RSWC               ;LOAD WC WITH -1
2458  004672  012777  000001  174204            MOV     #1,@RSCS1               ;DO NO-OP FUNCTION
2459  004700  032777  000001  174176            BIT     #1,@RSCS1               ;DID GO BIT CLEAR
2460  004706  001401                            BEQ     .+4                     ;YES
2461  004710  104001                            HLT     !CS1                    ;GO BIT SHOULD BE CLEARED
2462  004712  022777  150600  174176            CMP     #150600,@RSDS           ;DID ERR BITS SET?
2463  004720  001401                            BEQ     .+4                     ;NO
2464  004722  104040                            HLT     !DS                     ;ERR BIT SHOULD BE 0
2465  004724  022777  177777  174156            CMP     #-1,@RSWC               ;DID WC MOVE?
2466  004732  001401                            BEQ     .+4                     ;NO
2467  004734  104010                            HLT     !WC                     ;WC SHOULD = 1777777
2468  004736  005777  174152                    TST     @RSDA                   ;DID DA MOV
2469  004742  001401                            BEQ     .+4                     ;NO
2470  004744  104004                            HLT     !DA                     ;DA SHOULD =0
2471  004746  023777  001102  174136            CMP     @#OBUFSV,@RSBA          ;DID BA MOVE
2472  004754  001401                            BEQ     .+4                     ;NO
2473  004756  104020                            HLT     !BA                     ;BA MOVED
2474  004760  036777  174176  174134            BIT     UNCMP,@RSAS             ;AS SHOULD BE SET
2475  004766  001001                            BNE     .+4                     ;IS IT?
2476  004770  104100                            HLT     !AS                     ;NO
2477  004772  022777  000007  174120            CMP     #7,@RSER                ;DID ER CHANGE?
2478  005000  001401                            BEQ     .+4                     ;NO
2479  005002  104002                            HLT     !ER                     ;ER SHOULD NOT CHANGE
```

```
2480                                    ;*****************************************************************************
2481                                    ;TEST 40                    TEST NO-OP FUNCTION CODE 21
2482                                    ;*****************************************************************************
2483   005004  104400                   TST40:  SCOPE
2484
2485   005006  104414                   NOOP21: CLRDK                           ;CLEAR ALL RS REG
2486   005010  012767  177777  017532           MOV     #177777,OUTBUF          ;DATA TO BE XFERED
2487   005016  013777  001102  174066           MOV     @#OBUFSV,@RSBA          ;SET UP CURRENT ADDRESS
2488   005024  012777  177777  174056           MOV     #-1,@RSWC               ;LOAD WC WITH -1
2489   005032  012777  000021  174044           MOV     #21,@RSCS1              ;DO NO-OP FUNCTION
2490   005040  032777  000001  174036           BIT     #1,@RSCS1               ;DID GO BIT CLEAR
2491   005046  001401                           BEQ     .+4                     ;YES
2492   005050  104001                           HLT     !CS1                    ;GO BIT SHOULD BE CLEARED
2493   005052  005777  174042                   TST     @RSER                   ;DID ANY ERRORS OCCURR?
2494   005056  001401                           BEQ     .+4                     ;NO
2495   005060  104002                           HLT     !ER                     ;ALL ERROR BITS SHOULD BE 0
2496   005062  022777  177777  174020           CMP     #-1,@RSWC               ;DID WC MOVE?
2497   005070  001401                           BEQ     .+4                     ;NO
2498   005072  104010                           HLT     !WC                     ;WC SHOULD = 1777777
2499   005074  005777  174014                   TST     @RSDA                   ;DID DA MOV
2500   005100  001401                           BEQ     .+4                     ;NO
2501   005102  104004                           HLT     !DA                     ;DA SHOULD =0
2502   005104  023777  001102  174000           CMP     @#OBUFSV,@RSBA          ;DID BA MOVE
2503   005112  001401                           BEQ     .+4                     ;NO
2504   005114  104020                           HLT     !BA                     ;BA MOVED
2505   005116  036777  174040  173776           BIT     UNCMP,@RSAS             ;AS SHOULD NOT SET ON
2506   005124  001401                           BEQ     .+4                     ;A NO-OP FUNCTION
2507   005126  104100                           HLT     !AS                     ;AS SET WHY?
2508   005130  022777  004220  173746           CMP     #4220,@RSCS1            ;IS CS1 CORRECT?
2509   005136  001401                           BEQ     .+4                     ;YES
2510   005140  104040                           HLT     !DS                     ;CS1 SHOULD = 4220
```

```
                                ;*****************************************************************************
                                ;TEST 41                    TEST NO-OP FUNCTION CODE 21 WITH ERROR BITS SET
                                ;*****************************************************************************
005142  104400                  TST41:  SCOPE

                        NNOP21: CLRDK                   ;CLEAR ALL RS REG
005144  104414                  MOV     #7,@RSER        ;LOAD ER
005146  012777  000007  173744  BIT     UNCMP,@RSAS     ;IS ATA BIT SET?
005154  036777  174002  173740  BNE     .+4             ;YES
005162  001001                  HLT     !AS             ;AS BIT SHOULD BE SET
005164  104100                  MOV     #177777,OUTBUF  ;DATA TO BE XFERED
005166  012767  177777  017354  MOV     @#OBUFSV,@RSBA  ;SET UP CURRENT ADDRESS
005174  013777  001102  173710  MOV     #-1,@RSWC       ;LOAD WC WITH -1
005202  012777  177777  173700  MOV     #21,@RSCS1      ;DO NO-OP FUNCTION
005210  012777  000021  173666  BIT     #1,@RSCS1       ;DID GO BIT CLEAR
005216  032777  000001  173660  BEQ     .+4             ;YES
005224  001401                  HLT     !CS1            ;GO BIT SHOULD BE CLEARED
005226  104001                  CMP     #150600,@RSDS   ;DID ERR BITS SET?
005230  022777  150600  173660  BEQ     .+4             ;NO
005236  001401                  HLT     !DS             ;ERR BIT SHOULD BE 0
005240  104040                  CMP     #-1,@RSWC       ;DID WC MOVE?
005242  022777  177777  173640  BEQ     .+4             ;NO
005250  001401                  HLT     !WC             ;WC SHOULD = 1777777
005252  104010                  TST     @RSDA           ;DID DA MOV
005254  005777  173634          BEQ     .+4             ;NO
005260  001401                  HLT     !DA             ;DA SHOULD =0
005262  104004                  CMP     @#OBUFSV,@RSBA  ;DID BA MOVE
005264  023777  001102  173620  BEQ     .+4             ;NO
005272  001401                  HLT     !BA             ;BA MOVED
005274  104020                  BIT     UNCMP,@RSAS     ;AS SHOULD BE SET
005276  036777  173660  173616  BNE     .+4             ;IS IT?
005304  001001                  HLT     !AS             ;NO
005306  104100                  CMP     #7,@RSER        ;DID ER CHANGE?
005310  022777  000007  173602  BEQ     .+4             ;NO
005316  001401                  HLT     !ER             ;ER SHOULD NOT CHANGE
005320  104002                  CMP     #104220,@RSCS1  ;IS CS1 CORRECT?
005322  022777  104220  173554  BEQ     .+4             ;YES
005330  001401                  HLT     !DS             ;CS1 SHOULD = 104220
005332  104040
```

```
                                ;*****************************************************************************
                                ;TEST 42                    TEST DRIVE CLEAR FUNCTION WITH ERRORS SET
                                ;*****************************************************************************
005334  104400                  TST42:  SCOPE
                                ;FIRST SET ALL R/W BITS IN DISK REG
                                ;DO DRIVE CLEAR-ALL R/W BITS SHOULD BE CLEARED

005336  104414                  DRCLR:  CLRDK                   ;CLEAR ALL RS REG
005340  012777  177777  173542          MOV     #177777,@RSWC   ;LOAD RSWC
005346  012777  177777  173540          MOV     #177777,@RSDA   ;SET ALL POSSIBLE
005354  012777  177017  173536          MOV     #177017,@RSER   ;BITS IN DISK REG
005362  012777  000070  173540          MOV     #70,@RSMR       ;SET THESE BITS
005370  012777  000011  173506          MOV     #11,@RSCS1      ;SET DRIVE CLEAR
005376  017700  173504                  MOV     @RSCS2,BAD      ;GET CS2 DATA
005402  042700  177640                  BIC     #177640,BAD     ;CLEAR JUNK
005406  016701  173544                  MOV     UNNUM,GOOD      ;GET DRIVE UNIT
005412  052701  000100                  BIS     #100,GOOD       ;SET IR BIT
005416  020100                          CMP     GOOD,BAD        ;IS UNIT # THE SAME
005420  001401                          BEQ     .+4             ;YES
005422  104000                          HLT                     ;UNIT # IN CS2 GOT MODIFIED
005424  005777  173464                  TST     @RSDA           ;DID DA CLEAR
005430  001401                          BEQ     .+4             ;YES
005432  104004                          HLT     !DA             ;DA SHOULD BE 0
005434  005777  173460                  TST     @RSER           ;DID ER CLEAR
005440  001401                          BEQ     .+4             ;YES
005442  104002                          HLT     !ER             ;ER SHOULD BE CLEARED
005444  017767  173460  173526          MOV     @RSMR,WORK      ;GET MR REG
005452  042767  177707  173520          BIC     #177707,WORK    ;CLEAR JUNK
005460  022767  000070  173512          CMP     #70,WORK        ;IS 70 STILL SET IN MR?
005466  001401                          BEQ     .+4             ;YES
005470  104220                          HLT     !MR             ;BITS 70 SHOULD NOT CLEAR
005472  022777  004210  173404          CMP     #4210,@RSCS1    ;DID THESE BITS CLEAR?
005500  001401                          BEQ     .+4             ;YES
005502  104001                          HLT     !CS1            ;CS1 SHOULD =4210
005504  036777  173452  173410          BIT     UNCMP,@RSAS     ;AS SHOULD NOT SET
005512  001401                          BEQ     .+4             ;ON A DRIVE CLEAR FUN
005514  104100                          HLT     !AS             ;WHY DID AS SET?
005516  022777  177777  173364          CMP     #177777,@RSWC   ;DID RSWC CHANGE?
005524  001401                          BEQ     .+4             ;NO
005526  104010                          HLT     !WC
```

```
                                        ;DO ONE WORD WRITE
                                        ;*****************************************************************
                                        ;TEST 43                    EXECUTE THE ONE WORD WRITE
                                        ;*****************************************************************
005530  104400                          TST43:  SCOPE

005532  104414                  WRTST:  CLRDK                           ;CLEAR ALL RS REG
005534  012767  177777  017006          MOV     #177777,OUTBUF          ;DATA TO BE X-FERED
005542  013777  001102  173342          MOV     @#OBUFSV,@RSBA          ;SET UP CURRENT ADDRESS
005550  012777  177777  173332          MOV     #-1,@RSWC               ;SET WORD COUNT TO -1
005556  012777  000060  173320  1$:     MOV     #60,@RSCS1              ;SET FUNCTION WITH NO GO BIT
005564  022777  177777  173316          CMP     #-1,@RSWC               ;DID WC MOVE?
005572  001401                          BEQ     .+4                     ;NO
005574  104010                          HLT     !WC                     ;WC MOVED
005576  023777  001102  173306          CMP     @#OBUFSV,@RSBA          ;DID RSBA MOVE?
005604  001401                          BEQ     .+4                     ;NO
005606  104020                          HLT     !BA                     ;BA MOVED
005610  052777  000001  173266          BIS     #BIT0,@RSCS1            ;SET GO BIT
005616  105777  173262          2$:     TSTB    @RSCS1                  ;TEST FOR RDY=0
005622  100001                          BPL     .+4                     ;RDY=0
005624  104001                          HLT     !CS1                    ;RDY SHOULD = 0
005626  004767  016424                  JSR     PC,WAITRY               ;WAIT FOR READY
005632  104001                          HLT     !CS1                    ;SHOULD = 260 RDY NEVER CAME UP
005634  022777  000001  173252          CMP     #1,@RSDA                ;IS RSDA CORRECT
005642  001401                          BEQ     .+4                     ;RSDA OK
005644  104004                          HLT     !DA                     ;SHOULD = 1 SHOULD INCREMENT
005646  022777  004260  173230  3$:     CMP     #4260,@RSCS1            ;IS ERROR FLAG SET?
005654  001401                          BEQ     .+4                     ;NO! X-FER OK
005656  104047                          HLT     !CS1!ER!DS!DA           ;ERROR DURING X-FER
005660  005777  173224          4$:     TST     @RSWC                   ;FETCH WORD COUNT
005664  001401                          BEQ     .+4                     ;WORD COUNT DID OVERFLOW
005666  104010                          HLT     !WC                     ;SHOULD = 0 FAILED TO INCREMENT
005670  022777  010600  173220          CMP     #10600,@RSDS            ;IS RSDS OK?
005676  001401                          BEQ     .+4                     ;YES
005700  104044                          HLT     !DS!DA                  ;NO
005702  016701  173250                  MOV     UNNUM,GOOD              ;GET UNIT #
005706  052701  000100                  BIS     #100,GOOD               ;SET IR BIT
005712  017709  173170                  MOV     @RSCS2,BAD              ;GET CS2
005716  020100                          CMP     GOOD,BAD                ;IS CS2 CORRECT?
005720  001401                          BEQ     .+4                     ;YES
005722  104000                          HLT                             ;BAD = CS2 GOOD IS CORRECT ANS
005724  017700  173162                  MOV     @RSBA,BAD               ;GET BA DATA
005730  013701  001102                  MOV     @#OBUFSV,GOOD           ;WHAT RSBA SHOULD EQUAL
005734  062701  000002                  ADD     #2,GOOD                 ;UPDATE OUTBUFFER
005740  020100                          CMP     GOOD,BAD                ;IS RSBA CORRECT
005742  001401                          BEQ     .+4                     ;YES
005744  104000                          HLT                             ;BA FAILED TO INCREMENT
005746  005777  173146                  TST     @RSER                   ;DID ANY ERRORS SET?
005752  001401                          BEQ     .+4                     ;NO
005754  104040                          HLT     !DS
```

```
                                        ;TEST READ FUNCTION
                                        ;
                                        ;*****************************************************************************
                                        ;TEST 44                    EXECUTE THE ONE WORD READ
                                        ;*****************************************************************************
005756  104400                          TST44:  SCOPE
                                        ;
005760  104414                          RDTST:  CLRDK                           ;CLEAR ALL RS REG
005762  005067  016562                          CLR     OUTBUF                  ;CLR TO READ INTO
005766  013777  001102  173116                  MOV     @#OBUFSV,@RSBA          ;SET UP CURRENT ADDRESS
005774  012777  177777  173106                  MOV     #-1,@RSWC               ;SET WORD COUNT TO -1
006002  012777  000071  173074          1$:     MOV     #71,@RSCS1              ;GO READ
006010  105777  173070                  2$:     TSTB    @RSCS1                  ;TEST FOR BUSY=1
006014  100001                                  BPL     .+4                     ;BUSY SET
006016  104001                                  HLT     !CS1                    ;BUSY NOT SET
006020  004767  016232                          JSR     PC,WAITRY               ;WAIT FOR READY
006024  104001                                  HLT     !CS1                    ;TIMEOUT RDY DID NOT SET
006026  022777  000001  173060                  CMP     #BIT0,@RSDA             ;WAS RSDA INCREMENTED BY 1
006034  001401                                  BEQ     .+4                     ;RSDA OK
006036  104046                                  HLT     !DA!ER!DS               ;RSDA SHOULD CONTAIN A 1
006040  022777  004270  173036          3$:     CMP     #4270,@RSCS1            ;IS ERROR FLAG SET?
006046  001401                                  BEQ     .+4                     ;NO! X-FER OK
006050  104043                                  HLT     !CS1!ER!DS              ;RSCS1 SHOULD = 270
006052  005777  173032                  4$:     TST     @RSWC                   ;TEST WC
006056  001401                                  BEQ     .+4                     ;WORD COUNT DID OVERFLOW
006060  104010                                  HLT     !WC                     ;SHOULD = 0
006062  016701  173070                          MOV     UNNUM,GOOD              ;GET CORRECT
006066  052701  000100                          BIS     #100,GOOD               ;ANS OF CS2
006072  017700  173010                          MOV     @RSCS2,BAD              ;GET CS2
006076  020100                                  CMP     GOOD,BAD                ;IS CS2 CORRECT?
006100  001401                                  BEQ     .+4                     ;YES
006102  104000                                  HLT                             ;GOOD = CORRECT ANS FOR CS2
006104  017700  173002                          MOV     @RSBA,BAD               ;FETCH CURRENT ADDRESS
006110  013701  001102                          MOV     @#OBUFSV,GOOD           ;WHAT RSBA SHOULD EQUAL
006114  062701  000002                          ADD     #2,GOOD                 ;UPDATE IT
006120  020001                                  CMP     BAD,GOOD                ;IS RSBA CORRECT
006122  001401                                  BEQ     .+4                     ;YES EXECUTE CONTINUE
006124  104000                                  HLT                             ;RSBA FAILED TO INCREMENT
006126  016700  016416                          MOV     OUTBUF,BAD              ;GET DATA READ FROM DISK
006132  012701  177777                          MOV     #-1,GOOD                ;GET CORRECT ANS
006136  020100                                  CMP     GOOD,BAD                ;IS OUTBUF CORRECT
006140  001401                                  BEQ     .+4                     ;YES
006142  104000                                  HLT                             ;GOOD=CORRECT ANS BAD=DATA READ FROM DISK
```

```
                                ;*******************************************************************************
                                ;TEST 45                    TEST WRITE CHECK
                                ;*******************************************************************************
006144  104400                  TST45:  SCOPE
                                ;DO A ONE WORD WRITE CHECK

                                ;* * *EXECUTE THE ONE WORD WRITE CHECK* * *

006146  104414                  WRCKT:  CLRDK                   ;CLEAR ALL RS REG
006150  012767  177777  016372          MOV     #177777,OUTBUF  ;DATA TO BE X-FERED
006156  013777  001102  172726          MOV     @#OBUFSV,@RSBA  ;SET UP CURRENT ADDRESS
006164  012777  177777  172716          MOV     #-1,@RSWC       ;SET WORD COUNT TO -1
006172  012777  000051  172704  1$:     MOV     #51,@RSCS1      ;GO WRITE CHECK
006200  105777  172700          2$:     TSTB    @RSCS1          ;TEST FOR READY
006204  100001                          BPL     .+4             ;NOT READY
006206  104001                          HLT     !CS1            ;BUSY FAILED TO SET
006210  004767  016042          RSWCWT: JSR     PC,WAITRY       ;WAIT FOR READY
006214  104001                          HLT     !CS1            ;BUSY FAILED TO CLEAR
006216  016701  172734                  MOV     UNNUM,GOOD      ;GET UNIT #
006222  052701  000100                  BIS     #100,GOOD       ;SET BIT IR
006226  017700  172654                  MOV     @RSCS2,BAD      ;GET CS2
006232  020100                          CMP     GOOD,BAD        ;IS CS2 CORRECT?
006234  001401                          BEQ     .+4             ;YES
006236  104000                          HLT                     ;GOOD = CORRECT ANS FOR CS2
006240  022777  004250  172636  2$:     CMP     #4250,@RSCS1    ;ANY ERRORS?
006246  001401                          BEQ     .+4             ;X-FER OK
006250  104046                          HLT     !DA!ER!DS       ;ERROR DUR X-FER
006252  022777  000001  172634  3$:     CMP     #BIT0,@RSDA     ;WAS DAR INCREMENTED BY 1
006260  001401                          BEQ     .+4             ;RSDA OK
006262  104004                          HLT     !DA             ;DAR SHOULD = 1
006264  005777  172620                  TST     @RSWC           ;TEST FOR OVERFLOW
006270  001401                          BEQ     .+4             ;WORD COUNT DID OVERFLOW
006272  104010                          HLT     !WC             ;SHOULD = 0
006274  017700  172612                  MOV     @RSBA,BAD       ;FETCH CURRENT ADDRESS
006300  013701  001102                  MOV     @#OBUFSV,GOOD   ;WHAT RSBA SHOULD EQUAL
006304  062701  000002                  ADD     #2,GOOD         ;UPDATE IT
006310  020001                          CMP     BAD,GOOD        ;IS RSBA CORRECT
006312  001401                          BEQ     .+4             ;YES EXECUTE CONTINUE
006314  104000                          HLT                     ;RSBA FAILED TO INCREMENT
```

```
2721  006316  016767  172560  172650  NXM:    MOV     OBUFSV,BPORTT   ;SAVE -B- PORT BUFFER
2722  006324  012767  024550  172550          MOV     #OUTBUF,OBUFSV  ;RESTORE OBUFSV
2723
2724                                  ;DESELECT THEN SELECT UNIT NUMBER IN RSCS2 CHECK TIMING
2725                                  ;*************************************************************************
2726                                  ;TEST 46                    DESELECT THEN SELECT UNIT NUMBER TIMING TEST
2727                                  ;*************************************************************************
2728  006332  104400                  TST46:  SCOPE
2729
2730  006334  104414                  UNITST: CLRDK                           ;CLEAR ALL RS REG
2731  006336  005004                          CLR     R4                      ;CLEAR R4
2732  006340  020467  172612                  CMP     R4,UNNUM                ;IS THIS CORRECT UNIT #?
2733  006344  001001                          BNE     3$                      ;NO THEN USE IT
2734  006346  005204                          INC     R4                      ;GET WRONG DRIVE
2735  006350  012767  177777  016172  3$:     MOV     #177777,OUTBUF          ;DATA TO BE X-FERED
2736  006356  013777  001102  172526          MOV     @#OBUFSV,@RSBA          ;SET UP CURRENT ADDRESS
2737  006364  012777  177777  172516          MOV     #-1,@RSWC               ;SET WORD COUNT TO -1
2738  006372  012703  000061                  MOV     #61,R3                  ;GET WRITE FUNCTION
2739  006376  016705  172554                  MOV     UNNUM,R5                ;GET CORRECT UNIT #
2740  006402  012701  172040                  MOV     #172040,R1              ;GET CS1 REG
2741  006406  010461  000010                  MOV     R4,10(R1)               ;LOAD WRONG UNIT # INTO CS2
2742  006412  000240                          NOP                             ;WAIT FOR DRIVE TO SETTLE
2743  006414  010561  000010                  MOV     R5,10(R1)               ;LOAD CORRECT UNIT #
2744  006420  010311          1$:     MOV     R3,(R1)                 ;LOAD FUNCTION IN CS1
2745  006422  004767  015630                  JSR     PC,WAITRY               ;WAIT FOR READY
2746  006426  104001                          HLT     !CS1                    ;SHOULD = 260 RDY NEVER CAME UP
2747  006430  022777  004260  172446          CMP     #4260,@RSCS1            ;IS ERROR FLAG SET?
2748  006436  001401                          BEQ     .+4                     ;NO! X-FER OK
2749  006440  104047                          HLT     !CS1!ER!DS!DA           ;ERROR DURING X-FER
2750  006442  022777  000001  172444          CMP     #1,@RSDA                ;IS RSDA CORRECT
2751  006450  001401                          BEQ     .+4                     ;RSDA OK
2752  006452  104004                          HLT     !DA                     ;SHOULD = 1 SHOULD INCREMENT
2753  006454  005777  172430                  TST     @RSWC                   ;FETCH WORD COUNT
2754  006460  001401                          BEQ     .+4                     ;WORD COUNT DID OVERFLOW
2755  006462  104010                          HLT     !WC                     ;SHOULD = 0 FAILED TO INCREMENT
2756  006464  022777  010600  172424          CMP     #10600,@RSDS            ;IS RSDS OK?
2757  006472  001401                          BEQ     .+4                     ;YES
2758  006474  104044                          HLT     !DS!DA                  ;NO
2759  006476  016701  172454                  MOV     UNNUM,GOOD              ;GET UNIT #
2760  006502  052701  000100                  BIS     #100,GOOD               ;SET IR BIT
2761  006506  017700  172374                  MOV     @RSCS2,BAD              ;GET CS2
2762  006512  020100                          CMP     GOOD,BAD                ;IS CS2 CORRECT?
2763  006514  001401                          BEQ     .+4                     ;YES
2764  006516  104000                          HLT                             ;BAD = CS2 GOOD IS CORRECT ANS
2765  006520  017700  172366                  MOV     @RSBA,BAD               ;GET BA DATA
2766  006524  013701  001102                  MOV     @#OBUFSV,GOOD           ;WHAT RSBA SHOULD EQUAL
2767  006530  062701  000002                  ADD     #2,GOOD                 ;UPDATE OUTBUFFER
2768  006534  020100                          CMP     GOOD,BAD                ;IS RSBA CORRECT
2769  006536  001401                          BEQ     .+4                     ;YES
2770  006540  104000                          HLT                             ;BA FAILED TO INCREMENT
```

```
2771                                    ;TEST CURRENT ADDRESS INHIBT-BAI IN RSCS2
2772                                    ;DO A ONE WORD WRITE AND SEE
2773                                    ;IF RSBA INCREMENTED AFTER THE X-FER
2774
2775                                    ;*************************************************************************
2776                                    ;TEST 47                TEST BAI IN RSCS2
2777                                    ;*************************************************************************
2778  006542  104400                   TST47:  SCOPE
2779
2780  006544  104414                   BAITST: CLRDK                           ;CLEAR ALL RS REG
2781  006546  013777  001102  172336           MOV     @#OBUFSV,@RSBA          ;SET UP CURRENT ADDR
2782  006554  012777  177777  172326           MOV     #-1,@RSWC               ;SET WORD COUNT TO -1
2783  006562  052777  000010  172316           BIS     #BAI,@RSCS2             ;SET BAI BIT
2784  006570  012777  000061  172306           MOV     #61,@RSCS1              ;WRITE
2785  006576  004767  015454                   JSR     PC,WAITRY               ;WAIT FOR READY
2786  006602  104001                           HLT     !CS1                    ;RDY DID NOT SET
2787  006604  013701  001102           1$:     MOV     @#OBUFSV,GOOD           ;WHAT RSBA SHOULD BE
2788  006610  017700  172276                   MOV     @RSBA,BAD               ;WHAT RSBA IS
2789  006614  020100                           CMP     GOOD,BAD                ;COMPARE
2790  006616  001401                           BEQ     .+4                     ;YES
2791  006620  104000                           HLT                             ;BAD=OUTBUF GOOD = CORRECT ANS
2792  006622  005777  172272                   TST     @RSER                   ;ANY ERRORS?
2793  006626  001401                           BEQ     .+4                     ;NO
2794  006630  104040                           HLT     !DS                     ;YES
2795  006632  104414                           CLRDK                           ;CLEAR ALL RS REG
2796  006634  032777  000010  172244           BIT     #BAI,@RSCS2             ;DID BAI CLEAR?
2797  006642  001401                           BEQ     .+4                     ;YES
2798  006644  104002                           HLT     !ER                     ;BAI DID NOT SET
```

```
                        ;*****************************************************************************
                        ;TEST 50                    TEST NON-EXISTENT MEMORY ERROR BIT IN CS2
                        ;*****************************************************************************
006646  104400           TST50:  SCOPE

006650  104414           NXMTSM: CLRDK                           ;CLEAR ALL RS REG
006652  052777  000010  172226          BIS     #BAI,@RSCS2      ;SET BAI BIT
006660  012777  177777  172222          MOV     #-1,@RSWC        ;SET UP WORD COUNT
006666  012777  173000  172216          MOV     #173000,@RSBA    ;SET UP CURRENT ADDRESS
006674  012777  000077  172232          MOV     #77,@RSBAE       ;AND EXTENDED BITS
006702  012777  001471  172174          MOV     #1471,@RSCS1     ;READ AND LOAD A16 +A17 FOR 18 BIT ADDRESS
006710  004767  015342                  JSR     PC,WAITRY        ;WAIT FOR READY
006714  104040                          HLT     !DS              ;READY NEVER CAME UP
006716  016701  172234  TSTNEM: MOV     UNNUM,GOOD               ;GET UNIT NO.
006722  052701  004310          BIS     #4310,GOOD               ;SET BAI+OR BITS
006726  017700  172154          MOV     @RSCS2,BAD               ;GET CS2
006732  020100                  CMP     GOOD,BAD                 ;IS CS2 CORRECT?
006734  001401                  BEQ     .+4                      ;YES
006736  104000                  HLT                              ;BAD=CS2 GOOD=CORRECT ANS FOR CS2
006740  022777  145670  172136          CMP     #145670,@RSCS1   ;DID TRE SET?
006746  001401                  BEQ     .+4                      ;YES
006750  104001                  HLT     !CS1                     ;TRE SHOULD SET BECAUSE OF NEM
006752  012777  040000  172124          MOV     #TRE,@RSCS1      ;CLEAR TRE
006760  017700  172122          MOV     @RSCS2,BAD               ;GET CS2
006764  016701  172166          MOV     UNNUM,GOOD               ;GET DRIVE
006770  052701  000310          BIS     #310,GOOD                ;SET IR
006774  020100                  CMP     GOOD,BAD                 ;IS CS2 CORRECT?
006776  001401                  BEQ     .+4                      ;YES
007000  104200                  HLT     !CS2                     ;CS2=BAD GOOD IS CORRECT ANS FOR CS2

                        ;*****************************************************************************
                        ;TEST 51                    TEST BLOCK SEARCH FUNCTION, PIP AND DRY BIT AND ADDR. CONF BIT
                        ;*****************************************************************************
007002  104400           TST51:  SCOPE

007004  104414           BLOCK:  CLRDK                           ;CLEAR ALL RS REG
007006  012777  000032  172100          MOV     #32,@RSDA        ;DO A SEARCH FOR SECTOR 32
007014  013777  001102  172070          MOV     @#OBUFSV,@RSBA   ;LOAD REGS. TO MAKE
007022  012777  177777  172060          MOV     #-1,@RSWC        ;SURE THEY DO NOT CHANGE
007030  005067  172144                  CLR     WORK             ;SETUP FOR TIMEOUT ROUTINE
007034  032777  001000  172066  4$:     BIT     #1000,@RSMR      ;WAIT FOR DISK TO
007042  001004                  BNE     3$                       ;REACH SECTOR 32
007044  005367  172130          DEC     WORK                     ;TIME OUT
007050  001371                  BNE     4$                       ;ROUTINE
007052  104220                  HLT     !MR                      ;COULD NOT FIND SECTOR 32
```

```
2844  007054  005077  172034          3$:   CLR   @RSDA          ;NOW SEARCH FOR 0
2845  007060  012777  000031  172016        MOV   #31,@RSCS1      ;DO A BLOCK SEARCH FUNCTION
2846  007066  032777  000200  172022        BIT   #DRY,@RSDS      ;IS DRY CLEARED?
2847  007074  001402                        BEQ   1$              ;YES
2848  007076  104040                        HLT   !DS             ;DRY SHOULD BE CLEARED DURING A BLOCK SEARCH
2849  007100  000500                        BR    OOUT            ;GET OUT BECAUSE OF TIMING
2850  007102  032777  020000  172006  1$:   BIT   #20000,@RSDS    ;IS PIP SET?
2851  007110  001001                        BNE   .+4             ;YES
2852  007112  104040                        HLT   !DS             ;PIP SHOULD BE SET
2853  007114  012701  020000                MOV   #20000,GOOD     ;SETUP FOR TIMEOUT ROUTINE
2854  007120  005301                  2$:   DEC   GOOD            ;DO TIMEOUT
2855  007122  001466                        BEQ   TTMOUT          ;TIMED OUT
2856  007124  032777  000200  171764        BIT   #DRY,@RSDS      ;DID DRY SET?
2857  007132  001772                        BEQ   2$              ;NO
2858  007134  022777  110600  171754        CMP   #110600,@RSDS   ;DID PIP CLEAR?
2859  007142  001401                        BEQ   .+4             ;YES
2860  007144  104040                        HLT   !DS             ;PIP BIT DID NOT CLEAR
2861  007146  022777  104230  171730        CMP   #104230,@RSCS1  ;DID SC SET?
2862  007154  001401                        BEQ   .+4             ;YES
2863  007156  104041                        HLT   !CS1!DS         ;SC DID NOT SET
2864  007160  016767  171772  172012        MOV   UNNUM,WORK      ;GET CORRECT AS BIT
2865  007166  005001                        CLR   GOOD            ;IN RSAS REG
2866  007170  000261                        SEC                   ;THAT SHOULD
2867  007172  006101                  5$:   ROL   GOOD            ;BE SET
2868  007174  005767  172000              TST   WORK
2869  007200  001403                        BEQ   6$
2870  007202  005367  171772              DEC   WORK
2871  007206  000771                        BR    5$
2872  007210  020177  171706          6$:   CMP   GOOD,@RSAS      ;IS RSAS CORRECT?
2873  007214  001403                        BEQ   7$              ;YES
2874  007216  017700  171700              MOV   @RSAS,BAD       ;NO
2875  007222  104000                        HLT
2876  007224  010177  171672          7$:   MOV   GOOD,@RSAS      ;CLEAR AS REG
2877  007230  005777  171666              TST   @RSAS           ;DID IT CLEAR?
2878  007234  001401                        BEQ   .+4             ;YES
2879  007236  104100                        HLT   !AS             ;NO
2880  007240  022777  010600  171650        CMP   #10600,@RSDS    ;DID ATA CLEAR?
2881  007246  001401                        BEQ   .+4             ;YES
2882  007250  104040                        HLT   !DS             ;NO
2883  007252  023777  001102  171632        CMP   @#OBUFSV,@RSBA  ;DID BA MOVE?
2884  007260  001401                        BEQ   .+4             ;NO
2885  007262  104021                        HLT   !CS1!BA         ;BA MOVED WHY?
2886  007264  022777  177777  171616        CMP   #-1,@RSWC       ;DID WC MOVE?
2887  007272  001401                        BEQ   .+4             ;NO
2888  007274  104010                        HLT   !WC             ;WC MOVED WHY?
2889  007276  000401                        BR    OOUT            ;DONE GET OUT
2890  007300  104040              TTMOUT:   HLT   !DS             ;DYR NEVER CAME UP
2891  007302                    OOUT:                             ;DONE
```

```
                                ;*****************************************************************************
                                ;TEST 52                    ILLEGAL FUNCTION CODE TEST CODE 3 TO 51
                                ;*****************************************************************************
007302  104400                  TST52:  SCOPE

                                ;TEST ILF BIT IN RSER AND ERR BIT IN RSDS
                                ;ALSO CHECKS TO SEE IF WC,BA, OR DA GOT MODIFIED
                                ;IF WISHING TO LOOP ON ONE FUNCTION ONLY, LOAD
                                ;FUNCTION INTO LOCATION ILLTAB: AND 0 IN FOLLOWING LOCATION

007304  016767  011156  171656  ILL51:  MOV     TIMES,TIMSV     ;SAVE LOOP COUNT
007312  012767  000010  011146          MOV     #10,TIMES       ;LOOP TEN TIMES
007320  104414                        CLRDK                   ;CLEAR ALL RS REG
007322  012703  024464          1$:     MOV     #ILLTAB,R3      ;GET STARTING ADD OF TABLE
007326  012300                  3$:     MOV     (R3)+,BAD       ;GET ILL FUN
007330  001513                        BEQ     ILFDN           ;DONE GET OUT
007332  013777  001102  171552        MOV     @#OBUFSV,@RSBA  ;SET UP REGS.
007340  012777  177777  171542        MOV     #-1,@RSWC       ;TO CHECK FOR CHANGE
007346  010077  171532          2$:     MOV     BAD,@RSCS1      ;DO ILLEGAL FUNCTION
007352  042700  000001                BIC     #BIT0,BAD       ;CLEAR GO BIT
007356  010001                        MOV     BAD,GOOD        ;MOV ILLEGAL FUN INTO GOOD
007360  105777  171520          6$:     TSTB    @RSCS1          ;RDY SET?
007364  100375                        BPL     6$              ;NO
007366  052701  104200                BIS     #104200,GOOD    ;SET ERROR BITS
007372  017700  171506          4$:     MOV     @RSCS1,BAD      ;PUT CS1 INTO BAD
007376  020100                        CMP     GOOD,BAD        ;IS CS1 CORRECT?
007400  001401                        BEQ     .+4             ;YES
007402  104000                        HLT                     ;GOOD IS WHAT CS1 SHOULD =BAD=CS1
007404  022777  000001  171506        CMP     #1,@RSER        ;DID ILF SET?
007412  001401                        BEQ     .+4             ;YES
007414  104043                        HLT     !CS1!ER!DS      ;ILF DID NOT SET
007416  022777  150600  171472        CMP     #150600,@RSDS   ;IS DS GOOD?
007424  001401                        BEQ     .+4             ;YES
007426  104043                        HLT     !CS1!ER!DS      ;EPR DID NOT SET
007430  017700  171452                MOV     @RSCS2,BAD      ;GET CS2
007434  016701  171516                MOV     UNNUM,GOOD      ;GET UNIT #
007440  052701  000100                BIS     #100,GOOD       ;SET IR BIT
007444  020100                        CMP     GOOD,BAD        ;IS CS2 CORRECT?
007446  001401                        BEQ     .+4             ;YES
007450  104000                        HLT                     ;GOOD = CORRECT ANS FOR CS2
007452  016701  171504                MOV     UNCMP,GOOD      ;GET CORRECT DRIVE
007456  042701  177400                BIC     #177400,GOOD    ;CLEAR UNWANTED BITS
007462  017700  171434                MOV     @RSAS,BAD       ;GET RSAS REG
007466  020001                        CMP     BAD,GOOD        ;DID CORRECT UNIT ANSWER?
007470  001401                        BEQ     .+4             ;YES
007472  104100                        HLT     !AS             ;NO WRONG DRIVE ANSWERED
```

```
2938  007474  023777  001102  171410          CMP     @#OBUFSV,@RSBA  ;DID BA MOVE
2939  007502  001401                          BEQ     .+4             ;NO
2940  007504  104021                          HLT     !CS1!BA         ;BA MOVED ON AN ILLEGAL FUNCTION
2941  007506  022777  177777  171374          CMP     #-1,@RSWC       ;DID WC MOVE?
2942  007514  001401                          BEQ     .+4             ;NO
2943  007516  104011                          HLT     !CS1!WC         ;WC MOVED
2944  007520  005777  171370                  TST     @RSDA           ;DID DA MOVE
2945  007524  001401                          BEQ     .+4             ;NO
2946  007526  104005                          HLT     !CS1!DA         ;DA MOVED
2947  007530  104414                          CLRDK                   ;CLEAR ALL ERRORS
2948  007532  005777  171362                  TST     @RSER           ;DID ERRORS CLEAR
2949  007536  001401                          BEQ     .+4             ;YES
2950  007540  104040                          HLT     !DS             ;ILF DID NOT CLEAR
2951  007542  022777  004200  171334          CMP     #4200,@RSCS1    ;DID ERRORS IN CS1 CLEAR
2952  007550  001401                          BEQ     .+4             ;YES
2953  007552  104040                          HLT     !DS
2954  007554  000167  177546                  J       3$              ;CONTINUE UNTIL DONE
2955  007560                          ILFDN:                          ;DONE WITH ILLEGAL FUNCTION TEST
2956                                  ;TEST ILF BIT IN RSER AND ERR BIT IN RSDS
2957                                  ;ALSO CHECKS TO SEE IF WC,BA, OR DA GOT MODIFIED
2958                                  ;IF WISHING TO LOOP ON ONE FUNCTION ONLY, LOAD
2959                                  ;FUNCTION INTO LOCATION ILFTB2: AND 0 IN FOLLOWING LOCATION
2960  007560  104414                  ILLFUN: CLRDK                   ;CLEAR ALL RS REG
2961  007562  012703  024526          1$:     MOV     #ILFTB2,R3      ;GET TABLE OF ILL FUNS.
2962  007566  012300                  3$:     MOV     (R3)+,BAD       ;GET ILL FUN
2963  007570  001402                          BEQ     2$              ;DO THIS FUNCTION
2964  007572  000167  000422                  JMP     ILFDNE          ;DONE GET OUT
2965  007576  013777  001102  171306  2$:     MOV     @#OBUFSV,@RSBA  ;SET UP REGS.
2966  007604  012777  177777  171276          MOV     #-1,@RSWC       ;TO CHECK FOR CHANGE
2967  007612  010067  171364                  MOV     BAD,WORK1       ;SHOULD WE TEST
2968  007616  042767  177707  171356          BIC     #177707,WORK1   ;BA AND WC
2969  007624  010077  171254                  MOV     BAD,@RSCS1      ;DO ILLEGAL FUNCTION
2970  007630  042700  000001          10$:    BIC     #BIT0,BAD       ;CLEAR GO BIT
2971  007634  010001                          MOV     BAD,GOOD        ;MOV ILLEGAL FUN INTO GOOD
2972  007636  105777  171242          6$:     TSTB    @RSCS1          ;RDY SET?
2973  007642  100375                          BPL     6$              ;NO
2974  007644  052701  144200                  BIS     #144200,GOOD    ;SET ERROR BITS
2975  007650  017700  171230                  MOV     @RSCS1,BAD      ;PUT CS1 INTO BAD
2976  007654  020100                          CMP     GOOD,BAD        ;IS CS1 CORRECT?
2977  007656  001401                          BEQ     .+4             ;YES
2978  007660  104000                          HLT                     ;GOOD IS WHAT CS1 SHOULD =BAD=CS1
2979  007662  022777  000001  171230          CMP     #1,@RSER        ;DID ILF SET?
2980  007670  001401                          BEQ     .+4             ;YES
2981  007672  104043                          HLT     !CS1!ER!DS      ;ILF DID NOT SET
2982  007674  022777  150600  171214          CMP     #150600,@RSDS   ;IS DS GOOD?
2983  007702  001401                          BEQ     .+4             ;YES
2984  007704  104043                          HLT     !CS1!ER!DS      ;ERR DID NOT SET
2985  007706  005777  171202                  TST     @RSDA           ;DID DA MOVE?
2986  007712  001401                          BEQ     .+4             ;NO
2987  007714  104005                          HLT     !CS1!DA         ;DA MOVED
2988  007716  022767  000060  171256          CMP     #60,WORK1       ;IS THIS AN ILL WRITE FUN?
2989  007724  001430                          BEQ     11$             ;YES
2990  007726  022767  000050  171246          CMP     #50,WORK1       ;IS THIS AN ILL WRITE CHECK FUN?
2991  007734  001455                          BEQ     7$              ;YES--BRANCH
2992                                  ;WORK1=70
2993  007736  017700  171144                  MOV     @RSCS2,BAD      ;GET CS2
```

```
 2994  007742  016701  171210                MOV     UNNUM,GOOD      ;GET UNIT #
 2995  007746  052701  001100                BIS     #1100,GOOD      ;SET IR BIT
 2996  007752  020100                        CMP     GOOD,BAD        ;IS CS2 CORRECT?
 2997  007754  001401                        BEQ     .+4             ;YES
 2998  007756  104000                        HLT                     ;GOOD = CORRECT ANS FOR CS2
 2999  007760  023777  001102  171124        CMP     @#OBUFSV,@RSBA  ;DID BA MOVE
 3000  007766  001401                        BEQ     .+4             ;NO
 3001  007770  104021                        HLT     !CS1!BA         ;BA MOVED ON AN ILLEGAL FUNCTION
 3002  007772  022777  177777  171110        CMP     #-1,@RSWC       ;DID WC MOVE?
 3003  010000  001401                        BEQ     .+4             ;NO
 3004  010002  104011                        HLT     !CS1!WC         ;WC MOVED
 3005  010004  000471                        BR      4$              ;CONTINUE UNTIL DONE
 3006                          ;WORK1=60
 3007  010006  017700  171074        11$:    MOV     @RSCS2,BAD      ;GET CS2
 3008  010012  016701  171140                MOV     UNNUM,GOOD      ;GET UNIT #
 3009  010016  052701  001300                BIS     #1300,GOOD      ;SET IR BIT
 3010  010022  020100                        CMP     GOOD,BAD        ;IS CS2 CORRECT?
 3011  010024  001401                        BEQ     .+4             ;YES
 3012  010026  104000                        HLT                     ;GOOD = CORRECT ANS FOR CS2
 3013  010030  013767  001102  171142        MOV     @#OBUFSV,WORK   ;GET BUFFER ADDR.
 3014  010036  062767  000002  171134        ADD     #2,WORK         ;UPDATE IT
 3015  010044  026777  171130  171040        CMP     WORK,@RSBA      ;DID BA MOVE
 3016  010052  001401                        BEQ     .+4             ;YES
 3017  010054  104021                        HLT     !CS1!BA         ;BA MOVED ON AN ILLEGAL FUNCTION
 3018  010056  022777  000000  171024        CMP     #0,@RSWC        ;DID WC MOVE?
 3019  010064  001401                        BEQ     .+4             ;YES
 3020  010066  104011                        HLT     !CS1!WC         ;BAD WC
 3021                          ;WORK1=50
 3022  010070  017700  171012        7$:     MOV     @RSCS2,BAD      ;GET CS2
 3023  010074  016701  171056                MOV     UNNUM,GOOD      ;GET UNIT #
 3024  010100  052701  001300                BIS     #1300,GOOD      ;SET IR BIT
 3025  010104  020100                        CMP     GOOD,BAD        ;IS CS2 CORRECT?
 3026  010106  001401                        BEQ     .+4             ;YES
 3027  010110  104000                        HLT                     ;GOOD = CORRECT ANS FOR CS2
 3028  010112  013767  001102  171060        MOV     @#OBUFSV,WORK   ;GET BUFFER ADDR.
 3029  010120  062767  000002  171052        ADD     #2,WORK         ;UPDATE IT
 3030  010126  022763  000057  177776        CMP     #57,-2(R3)      ;WRITE CHECK REV.?
 3031  010134  001003                        BNE     8$              ;NO--BRANCH
 3032  010136  162767  000004  171034        SUB     #4,WORK         ;YES--BUFFER ADDR. GOES BACKWARDS
 3033  010144  026777  171030  170740  8$:   CMP     WORK,@RSBA      ;DID BA MOVE
 3034  010152  001401                        BEQ     .+4             ;YES
 3035  010154  104021                        HLT     !CS1!BA         ;BA DIDN'T INCREMENT PROPERLY
 3036  010156  022777  000000  170724        CMP     #0,@RSWC        ;DID WC MOVE?
 3037  010164  001401                        BEQ     .+4             ;YES
 3038  010166  104011                        HLT     !CS1!WC         ;WC DIDN'T INCREMENT PROPERLY
 3039  010170  104414                4$:     CLRCK                   ;CLEAR ALL ERRORS
 3040  010172  022777  004200  170704        CMP     #4200,@RSCS1    ;DID ERRORS CLEAR
 3041  010200  001401                        BEQ     .+4             ;YES
 3042  010202  104040                        HLT     !DS             ;NO
 3043  010204  005777  170710                TST     @RSER           ;DID ERROR CLEAR
 3044  010210  001401                        BEQ     .+4             ;NO
 3045  010212  104040                        HLT     !DS             ;YES
 3046  010214  000167  177346                JMP     3$              ;CONTINUE UNTIL DONE
 3047  010220                        ILFDNE:                         ;DONE WITH ILLEGAL FUNCTION TEST
 3048                          ;*****************************************************************
 3049                          ;TEST 53                TEST ILLEGAL FUNCTION CODE 67
```

```
3050                                    ;*****************************************************************************
3051  010220  104400                    TST53:  SCOPE
3052
3053  010222  104414                    ILF67:  CLRDK                   ;CLEAR ALL RS REG
3054  010224  012777  177777  170656            MOV     #-1,@RSWC       ;SET WC TO -1
3055  010232  012767  024552  170740            MOV     #<OUTBUF+2!2>,WORK ;GET OUTBUF ADD.
3056  010240  016777  170734  170644            MOV     WORK,@RSBA      ;LOAD ADDR.
3057  010246  012777  000067  170630            MOV     #67,@RSCS1      ;DO FUNCTION 67
3058  010254  105777  170624            1$:     TSTB    @RSCS1          ;DONE YET?
3059  010260  100375                            BPL     1$              ;NO
3060  010262  017700  170624                    MOV     @RSBA,BAD       ;GET BA REG
3061  010266  012701  024550                    MOV     #<OUTBUF+2&^C3>,GOOD ;GET CORRECT ANS FOR RSBA
3062  010272  020100                            CMP     GOOD,BAD        ;IS RSBA CORRECT?
3063  010274  001401                            BEQ     .+4             ;YES
3064  010276  104000                            HLT                     ;BAD=RSBA GOOD=CORRECT ANS.
3065  010300  016701  170652                    MOV     UNNUM,GOOD      ;GET UNIT NUMBER
3066  010304  052701  001300                    BIS     #1300,GOOD      ;SET IR AND OR BITS
3067  010310  017700  170572                    MOV     @RSCS2,BAD      ;GET CS2
3068  010314  020100                            CMP     GOOD,BAD        ;IS CS2 CORRECT?
3069  010316  001401                            BEQ     .+4             ;YES
3070  010320  104000                            HLT                     ;BAD=CS2 GOOD=CORRECT ANS
3071  010322  022777  000001  170570            CMP     #1,@RSER        ;IS RSER CORRECT?
3072  010330  001401                            BEQ     .+4             ;YES
3073  010332  104002                            HLT     !ER             ;ER IS WRONG
3074  010334  104414                            CLRDK                   ;CLEAR ALL RS REG
3075  010336  005777  170556                    TST     @RSER           ;DID ERROR CLEAR
3076  010342  001401                            BEQ     .+4             ;YES
3077  010344  104040                            HLT     !DS             ;NO
3078  010346  012777  177700  170534            MOV     #-100,@RSWC     ;SET WC TO -100
3079  010354  012767  024566  170616            MOV     #OUTBUF+16!2,WORK ;GET OUTBUF ADD.
3080  010362  016777  170612  170522            MOV     WORK,@RSBA      ;LOAD ADDR
3081  010370  012777  000067  170506            MOV     #67,@RSCS1      ;DO FUNCTION 67
3082  010376  105777  170502            2$:     TSTB    @RSCS1          ;DONE YET?
3083  010402  100375                            BPL     2$              ;NO
3084  010404  012701  024546                    MOV     #24546,GOOD     ;IF PROG. IS MODIFIED THIS
3085                                            ;NUMBER WILL HAVE TO BE CHANGED.
3086  010410  017700  170476                    MOV     @RSBA,BAD       ;GET BA REG
3087  010414  020100                            CMP     GOOD,BAD        ;IS RSBA CORRECT?
3088  010416  001401                            BEQ     .+4             ;YES
3089  010420  104000                            HLT                     ;BAD=RSBA GOOD=CORRECT ANS.
3090  010422  016701  170530                    MOV     UNNUM,GOOD      ;GET UNIT NUMBER
3091  010426  052701  001200                    BIS     #1200,GOOD      ;SET OR BIT
3092  010432  017700  170450                    MOV     @RSCS2,BAD      ;GET CS2
3093  010436  020100                            CMP     GOOD,BAD        ;IS CS2 CORRECT?
3094  010440  001401                            BEQ     .+4             ;YES
3095  010442  104000                            HLT                     ;BAD=CS2 GOOD=CORRECT ANS
3096  010444  022777  000001  170446            CMP     #1,@RSER        ;IS RSER CORRECT?
3097  010452  001401                            BEQ     .+4             ;YES
3098  010454  104002                            HLT     !ER             ;ER IS WRONG
3099  010456  012777  040011  170420            MOV     #40011,@RSCS1   ;CLEAR ERRORS
3100  010464  022777  004210  170412            CMP     #4210,@RSCS1    ;DID THEY CLEAR IN CS1
3101  010472  001401                            BEQ     .+4             ;YES
3102  010474  104040                            HLT     !DS             ;NO
3103  010476  005777  170416                    TST     @RSER           ;DID RSER CLEAR
3104  010502  001401                            BEQ     .+4             ;YES
3105  010504  104040                            HLT     !DS             ;NO
```

```
 3106  010506  005777  170410          TST     @RSAS           ;DID RSAS CLEAR
 3107  010512  001401                  BEQ     .+4             ;YES
 3108  010514  104100                  HLT     !AS             ;NO
```

```
3109
3110                                    ;*************************************************************************
3111                                    ;TEST 54                    TEST PAR IN RSER
3112                                    ;*************************************************************************
3113  010516  104400                    TST54:  SCOPE
3114
3115  010520  016767  170444  007740    PARTST: MOV     TIMSV,TIMES     ;RESTORE LOOP #
3116  010526  104414                            CLRDK                   ;CLEAR ALL RS REG
3117  010530  012777  000010  170362    1$:     MOV     #10,@RSER       ;SET PAR
3118  010536  022777  150600  170352            CMP     #150600,@RSDS   ;DID ERR,ATA AND DRY SET?
3119  010544  001401                            BEQ     .+4             ;YES
3120  010546  104042                            HLT     !DS!ER          ;ER SHOULD SET IF PAR SETS IN RSER
3121  010550  104414                            CLRDK                   ;CLEAR ALL RS REG
3122  010552  005777  170342                    TST     @RSER           ;DID PAR CLEAR?
3123  010556  001401                            BEQ     .+4             ;YES
3124  010560  104002                            HLT     !ER             ;PAR DID NOT CLEAR BY CLEAR  BIT
3125  010562  022777  010600  170326            CMP     #10600,@RSDS    ;DID ERROR BITS CLEAR
3126  010570  001401                            BEQ     .+4             ;YES
3127  010572  104040                            HLT     !DS             ;NO
```

```
                                    ;CHECK BITS 12 TO 15 FOR 0
                                    ;CHECK SECTOR FRACTION TO WATCH FOR MOVEMENT
                                    ;CHECK CS BITS IN LA AND ADDRESS CONFIRM IN MR REG

                                    ;*************************************************************************
                                    ;TEST 55                    LOOK AHEAD TEST
                                    ;*************************************************************************
010574  104400                      TST55:  SCOPE

010576  032777  170000  170320      LATST:  BIT     #170000,@RSLA   ;ARE BITS 12 TO 15 CLEARED?
010604  001401                              BEQ     .+4             ;YES
010606  104204                              HLT     !LA             ;BITS 12 TO 15 SHOULD BE CLEARED
010610  104400                      TST56:  SCOPE

                                    ;NOW TEST MOVEMENT IN SF BITS

010612  012767  171005  170360              MOV     #171005,WORK    ;SET UP FOR TIME OUT ROUTIME
010620  017701  170300                      MOV     @RSLA,GOOD      ;GET READING FROM LA
010624  042701  007700                      BIC     #7700,GOOD      ;GET RID OF CS BITS
010630  005367  170344              1$:     DEC     WORK            ;WAIT FOR DISK
010634  001407                              BEQ     ERRR            ;TYPE ERROR
010636  017700  170262                      MOV     @RSLA,BAD       ;READ LA
010642  042700  007700                      BIC     #7700,BAD       ;CLEAR CS BITS
010646  020100                              CMP     GOOD,BAD        ;DID SF BITS CHANGE?
010650  001767                              BEQ     1$              ;WAIT FOR TIME OUT
010652  000422                              BR      LATDON          ;LA OK CONT
010654                          ERRR:
010654  104402  010660                      TYPE    .+2             ;.ASCIZ <15><12>"SECTOR FRACTIONS NOT MOVING"
010716  104204                              HLT     !LA             ;TYPE LOOK AHEAD REG
010720                          LATDON:                             ;DONE CONT.
```

```
                                ;*************************************************************************
                                ;TEST 57                  CHECK CS BITS TO INCREMENT AND  ADDRESS CONFIRM BIT IN MR
                                ;*************************************************************************
010720  104400                  TST57:  SCOPE

010722  016767  007540  170240  CSTST:  MOV     TIMES,TIMSV     ;SAVE LOOP CT
010730  012767  000010  007530          MOV     #10,TIMES       ;LOOP 10 TIMES
010736  104414                        CLRDK                   ;CLEAR ALL RS REG.
010740  012701  001000                MOV     #1000,GOOD      ;LOAD COUNTER
010744  032777  001000  170156        BIT     #BIT9,@RSMR     ;IS ADD CONFIRM BIT 0?
010752  001407                        BEQ     ADDCF           ;YES CONTINUE
010754  005301                        DEC     GOOD            ;WAIT FOR
010756  001376                        BNE     .-2             ;DISK TO MOVE
010760  032777  001000  170142        BIT     #BIT9,@RSMR     ;IS AADD. CON. BIT BIT 0?
010766  001401                        BEQ     .+4             ;YES
010770  104220                        HLT     !MR             ;ADD. CONF. BIT ALWAYS A 1

                                ;NOW TEST TA BITS AND ADD. CON. BIT IN MR

010772  012777  177777  170114  ADDCF:  MOV     #-1,@RSDA       ;INIT RSDA
011000  012767  177777  170172  1$:     MOV     #-1,WORK        ;SETUP TIMEOUT COUNTER
011006  005277  170102                  INC     @RSDA           ;GET NEXT SECTOR
011012  022777  010000  170074          CMP     #10000,@RSDA    ;DONE ALL YET?
011020  001433                        BEQ     DONCS           ;YES
011022  005367  170152          2$:     DEC     WORK            ;DO TIMEOUT ROUTINE
011026  001427                        BEQ     TMEOUT          ;ADD. CON. NEVER CAME UP
011030  032777  001000  170072        BIT     #1000,@RSMR     ;DID ADD CONFIRM BIT SET?
011036  001771                        BEQ     2$              ;YES
011040  017700  170060                MOV     @RSLA,BAD       ;GET LA
011044  042700  000077                BIC     #77,BAD         ;CLEAR SF BITS
011050  012767  000006  170124        MOV     #6,WORK1        ;SET UP COUNTER
011056  006000                3$:     ROR     BAD             ;MOV SA BITS RIGHT
011060  005367  170116                DEC     WORK1           ;DO 6 TIMES
011064  001374                        BNE     3$              ;DONE YET?
011066  017701  170022                MOV     @RSDA,GOOD      ;GET DA
011072  042701  177700                BIC     #177700,GOOD    ;CLEAR JUNK
011076  020100                        CMP     GOOD,BAD        ;ARE SA BITS = IN DA AND LA REG.?
011100  001401                        BEQ     .+4             ;OK
011102  104000                        HLT                     ;GOOD =DA BAD = LA
011104  000735                        BR      1$              ;NO WAIT

011106  104262                  TMEOUT: HLT     !MR!ER!DS       ;ADDRESS CONFIRM BIT NEVER SET COULD BE BAD OR
                                                                ;BAD LA OR BAD COMPARE BETWEEN LA AND DA
011110                          DONCS:                          ;TEST DONE CONTINUE
```

```
                                ;*****************************************************************************
                                ;TEST 60                    PARITY TEST
                                ;*****************************************************************************
011110  104400                  TST60:  SCOPE

011112  016767  170052  007346  PART:   MOV     TIMSV,TIMES     ;RESTORE LOOP COUNTER
011120  012737  011300  000004          MOV     #PRTP,@#4       ;SETUP TIME OUT VECTOR
011126  012737  000340  000006          MOV     #340,@#6
011134  012702  172100                  MOV     #MPR0,R2        ;GET PAR REG
011140  005712                TSTAGN:   TST     (R2)            ;DOES IT EXIST
011142  012712  000004                  MOV     #WWP,@R2        ;YES SET WRITE WRONG PARITY
011146  012701  001200                  MOV     #WORK,R1        ;GET TEST LOCATION
011152  011111                1$:       MOV     @R1,@R1         ;WRITE WRONG PARITY
011154  005711                          TST     @R1             ;READ IT
011156  005712                          TST     @R2             ;DID PARITY ERROR SET?
011160  100402                          BMI     2$              ;YES
011162  005012                          CLR     @R2             ;CLEAR PARITY REG
011164  000446                          BR      PRTP1           ;GET NEXT PARITY REG
011166  012712  000001          2$:     MOV     #1,@R2          ;CLEAR WWP AND ENABLE BAD PARITY
011172  104414                          CLRDK                   ;CLEAR ALL RS REG
011174  012767  177777  013346          MOV     #177777,OUTBUF  ;DATA TO BE X-FERED
011202  012777  001200  167702          MOV     #WORK,@RSBA     ;SET UP CURRENT ADDRESS
011210  012777  177777  167672          MOV     #-1,@RSWC       ;SET WORD COUNT TO -1
011216  012777  000061  167660          MOV     #61,@RSCS1      ;GO WRITE
011224  105777  167654          3$:     TSTB    @RSCS1          ;DONE YET?
011230  100375                          BPL     3$              ;NO WAIT
011232  005012                          CLR     @R2             ;CLEAR PARITY REG
011234  005067  167740                  CLR     WORK            ;WRITE GOOD PARITY
011240  017700  167642                  MOV     @RSCS2,BAD      ;GET CS2
011244  012701  020100                  MOV     #20100,GOOD
011250  056701  167702                  BIS     UNNUM,GOOD      ;GET CORRECT AND FOR CS2
011254  020100                          CMP     GOOD,BAD        ;IS CS2 CORRECT?
011256  001401                          BEQ     .+4             ;YES
011260  104000                          HLT                     ;CS2 SHOULD = GOOD
011262  022777  144260  167614          CMP     #144260,@RSCS1  ;IS CS1 CORRECT?
011270  001401                          BEQ     .+4             ;YES
011272  104040                          HLT     !DS
011274  000167  000020                  JMP     NOPAR           ;GET OUT

                                ;TRAPOUT ROUTINE

011300  022626                  PRTP:   CMP     (6)+,(6)+       ;CLEAR STACK
011302  022702  172136          PRTP1:  CMP     #172136,R2      ;DONE YET?
011306  001404                          BEQ     NOPAR           ;YES NO PAR REG
011310  062702  000002                  ADD     #2,R2           ;NO TRY AGAIN
011314  000167  177620                  JMP     TSTAGN          ;RETRY

011320  012737  000006  000004  NOPAR:  MOV     #6,@#4
011326  005037  000006                  CLR     @#6
```

```
                                ;*****************************************************************************
                                ;TEST 61                    TEST WRITE CHECK ERROR
                                ;*****************************************************************************
011332  104400                  TST61:  SCOPE

                                ;WRITE A WORD OF 0 AND FLOAT A 1 TRSOUGH IT TO CAUSE WCE
                                ;SET BIT14 IN ONCEE AND WRITE A WD OF -1 AND FLOAT 0
                                ;TO CAUSE WCE

011334  104414                  WCETST: CLRDK                           ;CLEAR ALL RS REG
011336  005067  013206                  CLR     OUTBUF                  ;WRITE A WD OF 0
011342  013777  001102  167542  WCETT:  MOV     @#OBUFSV,@RSBA          ;SET UP CURRENT ADDRESS
011350  012777  177777  167532          MOV     #-1,@RSWC               ;SET WORD COUNT TO -1
011356  012777  000061  167520          MOV     #61,@RSCS1              ;GO WRITE
011364  105777  167514          3$:     TSTB    @RSCS1                  ;DONE YET?
011370  100375                          BPL     3$                      ;NO WAIT
011372  032767  040000  167564          BIT     #BIT14,ONCEE            ;WRITE A 1 OR 0?
011400  001410                          BEQ     2$                      ;WRITE A 0
011402  012767  177777  013140          MOV     #-1,OUTBUF              ;WRITE A 1
011410  000241                          CLC                             ;CLEAR CARRY
011412  006167  013132          6$:     ROL     OUTBUF                  ;FLOAT A 0 THROUGH BAD WD
011416  103123                          BCC     WCEDON                  ;DONE GET OUT
011420  000406                          BR      5$                      ;CHECK WCE
011422  005067  013122          2$:     CLR     OUTBUF                  ;WRITE A 0
011426  000261                          SEC                             ;SET CARRY
011430  006167  013114          1$:     ROL     OUTBUF                  ;FLOAT A 1
011434  103503                          BCS     WCEDNE                  ;GET OUT WHEN DONE
011436  013777  001102  167446  5$:     MOV     @#OBUFSV,@RSBA          ;SET UP CURRENT ADDRESS
011444  012777  177777  167436          MOV     #-1,@RSWC               ;SET WORD COUNT TO -1
011452  005077  167436                  CLR     @RSDA
011456  012777  000051  167420          MOV     #51,@RSCS1              ;GO WRITE CHECK
011464  105777  167414          4$:     TSTB    @RSCS1                  ;READY YET?
011470  100375                          BPL     4$                      ;NO WAIT
011472  017700  167410                  MOV     @RSCS2,BAD              ;GET CS2
011476  016701  167454                  MOV     UNNUM,GOOD              ;SET UNIT #
011502  052701  040300                  BIS     #40300,GOOD             ;SET BITS
011506  020100                          CMP     GOOD,BAD                ;IS CS2 CORRECT?
011510  001413                          BEQ     7$                      ;YES
011512  104000                          HLT                             ;BAD=CS2 GOOD=CORRECT ANS
011514  016700  013030                  MOV     OUTBUF,BAD              ;GET BAD WD THAT SHOULD CAUSE WCE
011520  005001                          CLR     GOOD                    ;GET GOOD WD IF WRITING 0
011522  032767  040000  167434          BIT     #BIT14,ONCEE            ;ARE WE WRITING 1 OR 0
011530  001402                          BEQ     8$                      ;0
011532  012701  177777                  MOV     #-1,GOOD                ;GET GOOD WD FOR 1
011536  104000                  8$:     HLT                             ;GOOD = CORRECT WD WRITTEN
                                                                        ;BAD = INCORRECT WD THAT WCE DID NOT CATCH
```

```
3297  011540  022777  144250  167336  7$:     CMP     #144250,@RSCS1  ;DID TRE SET?
3298  011546  001401                          BEQ     .+4             ;YES
3299  011550  104043                          HLT     !CS1!ER!DS      ;TRE SHOULD SET IF WCE SETS
3300  011552  017700  167334                  MOV     @RSBA,BAD       ;FETCH CURRENT ADDRESS
3301  011556  013701  001102                  MOV     @#OBUFSV,GOOD   ;WHAT RSBA SHOULD EQUAL
3302  011562  062701  000002                  ADD     #2,GOOD         ;UPDATE IT
3303  011566  020001                          CMP     BAD,GOOD        ;IS RSBA CORRECT
3304  011570  001401                          BEQ     .+4             ;YES EXECUTE CONTINUE
3305  011572  104000                          HLT                     ;RSBA FAILED TO INCREMENT
3306  011574  104414                          CLRDK                   ;CLEAR ALL RS REG
3307  011576  016701  167354                  MOV     UNNUM,GOOD      ;PUT DRIVE IN GOOD
3308  011602  052701  000100                  BIS     #100,GOOD       ;SET IR BIT
3309  011606  017700  167274                  MOV     @RSCS2,BAD      ;GET CS2
3310  011612  020100                          CMP     GOOD,BAD        ;IS CS2 CORRECT
3311  011614  001401                          BEQ     .+4             ;YES
3312  011616  104000                          HLT                     ;BAD =CS2 GOOD IS CORRECT ANS
3313  011620  022777  004200  167256          CMP     #4200,@RSCS1    ;DID TRE CLEAR?
3314  011626  001401                          BEQ     .+4             ;YES
3315  011630  104001                          HLT     !CS1            ;TRE DID NOT CLEAR WITH CLEAR
3316  011632  032767  040000  167324          BIT     #BIT14,ONCEE    ;FLOATION A 1 OR 0?
3317  011640  001673                          BEQ     1$              ;FLOAT 1
3318  011642  000663                          BR      6$              ;FLOAT 0
3319  011644  052767  040000  167312  WCEDNE: BIS     #BIT14,ONCEE    ;SET BIT14
3320  011652  104414                          CLRDK
3321  011654  012767  177777  012666          MOV     #-1,OUTBUFF
3322  011662  000167  177454                  JMP     WCETT           ;NOW WRITE -1 IN OUTBUF
3323  011666  042767  040000  167270  WCEDON: BIC     #BIT14,ONCEE    ;CLEAR TEST FLAG
```

```
3324                                    ;*****************************************************************************
3325                                    ;TEST 62                    TEST PROGRAM ERROR BIT IN RSCS2
3326                                    ;*****************************************************************************
3327  011674  104400                    TST62:  SCOPE
3328
3329  011676  104414                    PGETST: CLRDK                           ;CLEAR ALL RS REG
3330  011700  012767  177777  012642            MOV     #177777,OUTBUF          ;DATA TO BE X-FERED
3331  011706  013777  001102  167176            MOV     @#OBUFSV,@RSBA          ;SET UP CURRENT ADDRESS
3332  011714  012777  177000  167166            MOV     #177000,@RSWC           ;SET WORD COUNT
3333  011722  012777  000061  167154            MOV     #61,@RSCS1              ;GO WRITE
3334  011730  105777  167150          2$:       TSTB    @RSCS1                  ;IS RDY CLEARED YET?
3335  011734  100775                            BMI     2$                      ;NO WAIT
3336  011736  012777  000071  167140            MOV     #71,@RSCS1              ;GO READ
3337  011744  004767  012306                    JSR     PC,WAITRY               ;WAIT FOR READY
3338  011750  104001                            HLT     !CS1                    ;RDY NEVER CAME UP
3339  011752  022777  144260  167124            CMP     #144260,@RSCS1          ;IS CS1 CORRECT?
3340  011760  001401                            BEQ     .+4                     ;YES
3341  011762  104001                            HLT     !CS1                    ;TRE SHOULD SET BY SETTING PGE
3342  011764  005777  167132                    TST     @RSAS                   ;AS SHOULD = 0
3343  011770  001401                            BEQ     .+4                     ;YES
3344  011772  104100                            HLT     !AS                     ;RSAS SHOULD = 0
3345  011774  016701  167156                    MOV     UNNUM,GOOD              ;GET UNIT #
3346  012000  052701  002200                    BIS     #2200,GOOD              ;SET PGE, IR, AND OR
3347  012004  017700  167076                    MOV     @RSCS2,BAD              ;GET CS2
3348  012010  020100                            CMP     GOOD,BAD                ;IS IT CORRECT?
3349  012012  001401                            BEQ     .+4                     ;YES
3350  012014  104000                            HLT                             ;BAD = CS2
3351  012016  005777  167066                    TST     @RSWC                   ;SHOULD NOT BE 0
3352  012022  001001                            BNE     .+4                     ;BECAUSE PGE SHOULD ABORT
3353  012024  104011                            HLT     !WC!CS1                 ;CURRENT OPERATION
3354  012026  005777  167066                    TST     @RSER                   ;DID ANY ERRORS SET?
3355  012032  001401                            BEQ     .+4                     ;NO
3356  012034  104040                            HLT     !DS                     ;RMR SHOULD BE SET
3357  012036  052777  040000  167040            BIS     #TRE,@RSCS1             ;CLEAR ERRORS
3358  012044  042701  002000                    BIC     #PGE,GOOD               ;CLEAR PGE ERROR
3359  012050  017700  167032                    MOV     @RSCS2,BAD              ;GET CS2
3360  012054  020100                            CMP     GOOD,BAD                ;IS CS2 CORRECT?
3361  012056  001401                            BEQ     .+4                     ;YES
3362  012060  104200                            HLT     !CS2                    ;PGE DID NOT CLEAR BY CLEARING TRE BAD = CS2
3363  012062  022777  004260  167014            CMP     #4260,@RSCS1            ;DID SC CLEAR
3364  012070  001401                            BEQ     .+4                     ;YES
3365  012072  104040                            HLT     !DS                     ;DID NOT CLEAR BY CLEARING TRE
```

```
                                ;*****************************************************************************
                                ;TEST 63                    TEST RMR IN RSER REGISTER TRYING TO WRITE INTO RSDA
                                ;*****************************************************************************
012074  104400                TST63:  SCOPE

012076  104414                RMRT1:  CLRDK                           ;CLEAR ALL RS REG
012100  013777  001102  167004        MOV     @#OBUFSV,@RSBA          ;SET UP CURRENT ADDRESS
012106  012777  177700  166774        MOV     #177700,@RSWC           ;SET WORD COUNT
012114  012703  172060                MOV     #172060,R3              ;GET RSLA REG
012120  011304                1$:     MOV     @R3,R4                  ;WAIT FOR
012122  042704  000017                BIC     #17,R4                  ;THE MIDDLE
012126  022704  000020                CMP     #20,R4                  ;OF SECTOR 0
012132  001372                        BNE     1$                      ;BEFORE DOING A SEARCH
012134  012777  000031  166742        MOV     #31,@RSCS1              ;SEARCH
012142  012777  007777  166744        MOV     #7777,@RSDA             ;CAUSE ERROR
012150  004767  012102                JSR     PC,WAITRY               ;WAIT FOR READY
012154  104001                        HLT     !CS1                    ;RDY NEVER CAME UP
012156  022777  000004  166734        CMP     #4,@RSER                ;DID RMR SET?
012164  001401                        BEQ     .+4                     ;YES
012166  104002                        HLT     !ER                     ;ER SHOULD = 4
012170  022777  007777  166716        CMP     #7777,@RSDA             ;DID DA GET MODIFIED?
012176  001001                        BNE     .+4                     ;NO
012200  104004                        HLT     !DA
012202  022777  150600  166706        CMP     #150600,@RSDS           ;DID ERR SET?
012210  001401                        BEQ     .+4                     ;YES
012212  104042                        HLT     !DS!ER                  ;ER DID NOT SET BECAUSE OF RMR
012214  022777  104230  166662        CMP     #104230,@RSCS1          ;IS CS1 CORRECT?
012222  001401                        BEQ     .+4                     ;YES
012224  104040                        HLT     !DS                     ;CS1 SHOULD = 144260
012226  104414                        CLRDK                           ;CLEAR ALL RS REG
012230  005777  166664                TST     @RSER                   ;DID RMR CLEAR?
012234  001401                        BEQ     .+4                     ;YES
012236  104002                        HLT     !ER                     ;RMR DID NOT CLEAR WITH A CLEAR
012240  022777  004200  166636        CMP     #4200,@RSCS1            ;IS CS1 CORRECT?
012246  001401                        BEQ     .+4                     ;YES
012250  104040                        HLT     !DS                     ;NO

                                ;*****************************************************************************
                                ;TEST 64                    TEST RMR IN RSER REGISTER TRYING TO WRITE INTO RSER
                                ;*****************************************************************************
012252  104400                TST64:  SCOPE

012254  104414                RMRT2:  CLRDK                           ;CLEAR ALL RS REG
012256  013777  001102  166626        MOV     @#OBUFSV,@RSBA          ;SET UP CURRENT ADDRESS
012264  012777  177700  166616        MOV     #177700,@RSWC           ;SET WORD COUNT
012272  012777  000061  166604        MOV     #61,@RSCS1              ;GO WRITE
012300  105777  166600                TSTB    @RSCS1          2$:     ;IS RDY SET?
012304  100775                        BMI     2$                      ;YES WAIT FOR IT TO CLEAR
012306  012777  177773  166604        MOV     #177773,@RSER           ;CAUSE ERROR
```

```
3415  012314  004767  011736          JSR     PC,WAITRY
3416  012320  104001                  HLT     !CS1                    ;RDY NEVER CAME UP
3417  012322  022777  000004  166570  CMP     #4,@RSER                ;DID RMR SET?
3418  012330  001401                  BEQ     .+4                     ;YES
3419  012332  104002                  HLT     !ER                     ;ER SHOULD = 4
3420  012334  022777  150600  166554  CMP     #150600,@RSDS           ;DID ERR SET?
3421  012342  001401                  BEQ     .+4                     ;YES
3422  012344  104042                  HLT     !DS!ER                  ;ERR DID NOT SET BECAUSE OF RMR
3423  012346  022777  144260  166530  CMP     #144260,@RSCS1          ;IS CS1 CORRECT?
3424  012354  001401                  BEQ     .+4                     ;YES
3425  012356  104040                  HLT     !DS                     ;CS1 SHOULD = 144260
3426  012360  104414                  CLRDK                           ;CLEAR ALL RS REG
3427
3428                          ;*************************************************************************
3429                          ;TEST 65                TEST RMR IN RSER REGISTER TRYING TO WRITE INTO RSCS1
3430                          ;*************************************************************************
3431  012362  104400          TST65:  SCOPE
3432
3433  012364  104414          RMRT3:  CLRDK                           ;CLEAR ALL RS REG
3434  012366  013777  001102  166516  MOV     @#OBUFSV,@RSBA          ;SET UP CURRENT ADDRESS
3435  012374  012777  177700  166506  MOV     #177700,@RSWC           ;SET WORD COUNT
3436  012402  012777  000061  166474  MOV     #61,@RSCS1              ;GO WRITE
3437  012410  105777  166470  2$:     TSTB    @RSCS1                  ;IS RDY SET?
3438  012414  100775                  BMI     2$                      ;YES WAIT FOR IT TO CLEAR
3439  012416  012777  000030  166460  MOV     #30,@RSCS1              ;CAUSE ERROR
3440  012424  004767  011626          JSR     PC,WAITRY               ;WAIT FOR READY
3441  012430  104001                  HLT     !CS1                    ;RDY NEVER CAME UP
3442  012432  022777  000004  166460  CMP     #4,@RSER                ;DID RMR SET?
3443  012440  001401                  BEQ     .+4                     ;YES
3444  012442  104002                  HLT     !ER                     ;ER SHOULD = 4
3445  012444  022777  150600  166444  CMP     #150600,@RSDS           ;DID ERR SET?
3446  012452  001401                  BEQ     .+4                     ;YES
3447  012454  104042                  HLT     !DS!ER                  ;ERID NOT SET BECAUSE OF RMR
3448  012456  022777  144260  166420  CMP     #144260,@RSCS1          ;IS CS1 CORRECT?
3449  012464  001401                  BEQ     .+4                     ;YES
3450  012466  104040                  HLT     !DS                     ;CS1 SHOULD = 144260
3451  012470  104414                  CLRDK                           ;CLEAR ALL RS REG
```

```
                                ;*****************************************************************************
                                ;TEST 66                TEST THAT RMR DOES NOT SET BY WRITTING INTO RSAS
                                ;*****************************************************************************
012472  104400                  TST66:  SCOPE
                                
012474  104414                  RMRT4:  CLRDK                           ;CLEAR ALL RS REG
012476  013777  001102  166406          MOV     @#OBUFSV,@RSBA          ;SET UP CURRENT ADDRESS
012504  012777  177700  166376          MOV     #177700,@RSWC           ;SET WORD COUNT
012512  012703  172060                  MOV     #172060,R3              ;GET RSLA REG
012516  011304                  1$:     MOV     @R3,R4                  ;WAIT FOR
012520  042704  000017                  BIC     #17,R4                  ;THE MIDDLE
012524  022704  000020                  CMP     #20,R4                  ;OF SECTOR 0
012530  001372                          BNE     1$                      ;BEFORE DOING A SEARCH
012532  012777  000031  166344          MOV     #31,@RSCS1              ;SEARCH
012540  012777  000000  166354          MOV     #0,@RSAS                ;TRY TO CAUSE ERROR
012546  005067  166426                  CLR     WORK                    ;CLEAR COUNTER
012552  032777  000200  166336  2$:     BIT     #BIT7,@RSDS             ;WAIT FOR DRY
012560  001004                          BNE     3$                      ;READY CONT
012562  005267  166412                  INC     WORK                    ;COUNT
012566  001371                          BNE     2$                      ;RETRY
012570  104001                          HLT     !CS1                    ;RDY NEVER CAME UP
012572  005777  166322          3$:     TST     @RSER                   ;DID RMR SET?
012576  001401                          BEQ     .+4                     ;NO
012600  104002                          HLT     !ER                     ;ER SHOULD = 0
012602  022777  110600  166306          CMP     #110600,@RSDS           ;DID ERR SET?
012610  001401                          BEQ     .+4                     ;NO
012612  104042                          HLT     !DS!ER                  ;DS SHOULD = 110600
012614  022777  104230  166262          CMP     #104230,@RSCS1          ;IS CS1 CORRECT?
012622  001401                          BEQ     .+4                     ;YES
012624  104040                          HLT     !DS                     ;CS1 SHOULD = 144260
012626  104414                          CLRDK                           ;CLEAR ALL RS REG
012630  022777  004200  166246          CMP     #4200,@RSCS1            ;IS CS1 CORRECT?
012636  001401                          BEQ     .+4                     ;YES
012640  104040                          HLT     !DS                     ;NO
```

```
                                ;*****************************************************************************
                                ;TEST 67                    TEST DCK IN RSER
                                ;*****************************************************************************
012642  104400                  TST67:  SCOPE

                                ;DO A WRITE AND THEN A CLEAR FUNCTION THAT SHOULD CAUSE DCK TO SET

                        DCKTST: CLRDK                           ;CLEAR ALL RS REG
012644  012767  177500  166330          MOV     #177500,WORK2   ;GET WC FOR RS04
012654  005767  166306                  TST     RS04DT          ;IS THIS A RS04?
012660  001003                          BNE     1$              ;YES
012662  012767  177600  166314          MOV     #177600,WORK2   ;NO
012670  016777  166310  166212  1$:     MOV     WORK2,@RSWC     ;LOAD WC
012676  012767  177777  011644          MOV     #-1,OUTBUF      ;WRITE -1
012704  013777  001102  166200          MOV     @#OBUFSV,@RSBA  ;SET UP CURRENT ADDRESS
012712  052777  000010  166166  4$:     BIS     #10,@RSCS2      ;SET BAI BIT
012720  012777  000061  166156          MOV     #61,@RSCS1      ;GO WRITE
012726  105777  166152          5$:     TSTB    @RSCS1          ;IS RDY SET?
012732  100375                          BPL     5$              ;WAIT FOR WRITE TO FINISH
012734  005077  166154          2$:     CLR     @RSDA           ;SET DSK ADDRESS TO 0
012740  005067  011604                  CLR     OUTBUF          ;WRITE 0
012744  013777  001102  166140          MOV     @#OBUFSV,@RSBA  ;SET UP CURRENT ADDRESS
012752  012777  177777  166130          MOV     #-1,@RSWC       ;LOAD WC
012760  012702  172060                  MOV     #172060,R2      ;PUT RSLA ADDR INTO R2
012764  011203                  3$:     MOV     (R2),R3         ;GET LA AND WAIT FOR
012766  042703  000077                  BIC     #77,R3          ;SECTOR 40
012772  022703  004000                  CMP     #4000,R3        ;BEFORE
012776  001372                          BNE     3$              ;WRITING
013000  012777  000061  166076          MOV     #61,@RSCS1      ;GO WRITE
013006  011203                  6$:     MOV     (R2),R3         ;GET RSLA AND WAIT FOR
013010  042703  000017                  BIC     #17,R3          ;MIDDLE OF SECTOR
013014  022703  000020                  CMP     #20,R3          ;0 BEFORE EXECUTING
013020  001372                          BNE     6$              ;A CLEAR FUNCTION
013022  012777  000040  166056          MOV     #40,@RSCS2      ;CLEAR ALL REG. DO IT THIS WAY
013030  016777  166122  166050          MOV     UNNUM,@RSCS2    ;DO NOT USE TRAP
```

```
3521  013036  105777  166042          INCW:  TSTB   @RSCS1          ;IS BUSY CLEARED
3522  013042  100401                         BMI    6$              ;FLAG CLEARED
3523  013044  104001                         HLT    !CS1            ;RDY NEVER CAME UP
3524  013046  013777  001102  166036  6$:    MOV    @#OBUFSV,@RSBA  ;SET UP CURRENT ADDRESS
3525  013054  016777  166124  166026         MOV    WORK2,@RSWC     ;LOAD WC
3526  013062  012777  000071  166014         MOV    #71,@RSCS1      ;GO READ
3527  013070  105777  166010          5$:    TSTB   @RSCS1          ;IS RDY SET?
3528  013074  100375                         BPL    5$              ;WAIT FOR READ TO FINISH
3529  013076  022777  100000  166014         CMP    #100000,@RSER   ;DID DCK SET?
3530  013104  001401                         BEQ    .+4             ;YES
3531  013106  104002                         HLT    !ER             ;DCK DID NOT SET
3532  013110  022777  150600  166000         CMP    #150600,@RSDS   ;DID ERR SET?
3533  013116  001401                         BEQ    .+4             ;YES
3534  013120  104040                         HLT    !DS             ;ER DID NOT SET BY DCK
3535  013122  022777  144270  165754         CMP    #144270,@RSCS1  ;IS CS1 CORRECT?
3536  013130  001401                         BEQ    .+4             ;YES
3537  013132  104044                         HLT    !DS!DA
3538  013134  017700  165746                 MOV    @RSCS2,BAD      ;GET CS2
3539  013140  016701  166012                 MOV    UNNUM,GOOD      ;GET UNIT #
3540  013144  052701  000100                 BIS    #100,GOOD       ;SET IR
3541  013150  020100                         CMP    GOOD,BAD        ;IS CS2 CORRECT?
3542  013152  001401                         BEQ    .+4             ;YES
3543  013154  104000                         HLT
3544  013156  012701  177700                 MOV    #177700,GOOD    ;NO
3545  013162  017700  165722          1$:    MOV    @RSWC,BAD       ;DID TRANSFER STOP AT END OF SECTOR?
3546  013166  020100                         CMP    GOOD,BAD
3547  013170  001401                         BEQ    .+4             ;YES
3548  013172  104000                         HLT                    ;NO
3549  013174  012701  024550                 MOV    #OUTBUF,GOOD    ;GET BA
3550  013200  005767  165762                 TST    RS04DT          ;RS04?
3551  013204  001003                         BNE    2$              ;YES
3552  013206  062701  000200                 ADD    #200,GOOD       ;RS03
3553  013212  000402                         BR     3$
3554  013214  062701  000400          2$:    ADD    #400,GOOD       ;GET CORRECT ANS FOR BA
3555  013220  017700  165666          3$:    MOV    @RSBA,BAD       ;GET BA
3556  013224  020001                         CMP    BAD,GOOD        ;IS BA CORRECT?
3557  013226  001401                         BEQ    .+4             ;YES
3558  013230  104000                         HLT                    ;NO
3559  013232  104414                         CLRDK                  ;CLEAR ALL RS REG
3560  013234  005777  165660                 TST    @RSER           ;DID DCK CLEAR?
3561  013240  001401                         BEQ    .+4             ;YES
3562  013242  104002                         HLT    !ER             ;DCK DID NOT CLEAR WITH CLEAR
3563  013244  012777  177500  165636         MOV    #177500,@RSWC   ;CLEAR DCK ON
3564  013252  013777  001102  165632         MOV    @#OBUFSV,@RSBA  ;DRIVE BY WRITING
3565  013260  012777  000061  165616         MOV    #61,@RSCS1      ;GOOD DATA
3566  013266  105777  165612          4$:    TSTB   @RSCS1          ;ON DRIVE
3567  013272  100375                         BPL    4$
```

```
                                ;TEST THE ABILITY OF THE DISK CONTROL TO
                                ;INCREMENT THE TRACK REGISTER.
                                ;
                                ;A ONE WORD WRITE WILL BE EXECUTED
                                ;RSDA=7777 RSWC = -1
                                ;AT THE COMPLETION OF THE WRITE RSDA = 10000
                                ;*************************************************************************
                                ;TEST 70                 TEST DISK ADDRESS REGISTER
                                ;*************************************************************************
013274  104400                  TST70:  SCOPE
                                ;
013276  104414                  DKADR:  CLRDK                           ;CLEAR ALL RS REG
013300  012777  177777  165602          MOV     #177777,@RSWC           ;SET WORD COUNT TO -1
013306  013777  001102  165576          MOV     @#OBUFSV,@RSBA          ;SET UP CURRENT ADDRESS
013314  012777  007777  165572          MOV     #7777,@RSDA             ;SET RSDA TO ALL ONES
013322  012777  000061  165554          MOV     #61,@RSCS1              ;GO WRITE ONE WORD
013330  004767  010722                  JSR     PC,WAITRY               ;WAIT FOR READY
013334  104001                          HLT     !CS1                    ;RDY DID NOT COME UP
013336  027727  165552  010000  5$:     CMP     @RSDA,#10000            ;DOES RSDA=0
013344  001401                          BEQ     .+4                     ;RSDA OK
013346  104004                          HLT     !DA                     ;DA DID NOT INCREMENT
                                ;
                                ;*************************************************************************
                                ;TEST 71                 TEST IAE ERROR
                                ;*************************************************************************
013350  104400                  TST71:  SCOPE
                                ;
                                ;IAE ERROR SHOULD SET ERR,ATA AND SC BITS
                                ;
013352  104414                  IAERR:  CLRDK                           ;CLEAR ALL RS REG
013354  012777  177777  165526          MOV     #177777,@RSWC           ;SET WC TO -1
013362  013777  001102  165522          MOV     @#OBUFSV,@RSBA          ;SET UP BUS ADDRESS
013370  012777  017777  165516          MOV     #17777,@RSDA            ;SET DA TO RECEIVE ERROR
013376  012777  000061  165500          MOV     #61,@RSCS1              ;GO WRITE ONE WD
013404  105777  165474          7$:     TSTB    @RSCS1                  ;TEST FOR ERR OR RDY
013410  100401                          BMI     .+4                     ;OK CONT.
013412  000774                          BR      7$                      ;WAIT
013414  022777  002000  165476          CMP     #2000,@RSER             ;DID IAE SET?
013422  001401                          BEQ     .+4                     ;YES
013424  104002                          HLT     !ER                     ;IAE SHOULD BE SET
013426  022777  150600  165462          CMP     #150600,@RSDS           ;DID ERR SET?
013434  001401                          BEQ     .+4                     ;YES
013436  104140                          HLT     !DS!AS                  ;ERR SHOULD BE SET
013440  022777  144260  165436          CMP     #144260,@RSCS1          ;DID SC SET?
013446  001401                          BEQ     .+4                     ;YES
013450  104001                          HLT     !CS1                    ;SC SHOULD BE SET
013452  104414                          CLRDK                           ;CLEAR ALL RS REG
013454  005777  165440                  TST     @RSER                   ;CLR ERRORS?
013460  001401                          BEQ     .+4                     ;YES
013462  104002                          HLT     !ER                     ;ERR DID NOT CLR WITH 40 IN CS2
```

```
                                        ;IN THIS ROUTINE THE PROGRAM WILL GENERATE A
                                        ;NON-EXISTENT DISK ERROR
                                        ;
                                        ;*************************************************************************
                                        ;TEST 72                TEST FOR NON=EXISTENT DISK ERROR
                                        ;*************************************************************************
013464  104400                          TST72:  SCOPE
                                        ;
013466  104414                          NEDTST: CLRDK                   ;CLEAR ALL RS REG
013470  012767  000401  165502                  MOV     #401,WORK       ;SET UP FOR N.E.D. NUMBER
013476  005001                                  CLR     GOOD            ;LOOK FOR NON EXISTENT DRIVES
013500  036767  165474  165452  1$:             BIT     WORK,UNITSV     ;ON THE SYSTEM
013506  001405                                  BEQ     3$              ;FOUND NON EXISTENT DRIVE
013510  005201                                  INC     GOOD            ;CONTAINS UNIT #
013512  006167  165462                          ROL     WORK            ;KEEP LOOKING
013516  103452                                  BCS     NEDDON          ;COULD NOT FIND ANY NON EXISTENT DRIVES
013520  000767                                  BR      1$              ;LOOK FOR NED
013522  010177  165360          3$:             MOV     GOOD,@RSCS2     ;LOAD NED IN CS2
013526  005077  165362                          CLR     @RSDA           ;WRITE DRIVE REG
013532  005777  165362                          TST     @RSER           ;DID ANY BITS SET IN RSER?
013536  001401                                  BEQ     .+4             ;NO
013540  104040                                  HLT     ;DS             ;WHY DID RSER CHANGE?
013542  017700  165340                          MOV     @RSCS2,BAD      ;GET CS2
013546  052701  010100                          BIS     #10100,GOOD     ;SET NED AND IR
013552  020100                                  CMP     GOOD,BAD        ;IS CS2 CORRECT?
013554  001401                                  BEQ     .+4             ;YES
013556  104000                                  HLT                     ;GOOD=CORRECT CS2 BAD=CS2
013560  022777  160200  165316                  CMP     #160200,@RSCS1  ;IS CS1 CORRECT?
013566  001401                                  BEQ     .+4             ;YES
013570  104200                                  HLT     ;CS2            ;TRE SHOULD SET BY NED ERROR
013572  005777  165324                          TST     @RSAS           ;DID ANY BITS SET?
013576  001401                                  BEQ     .+4             ;NO
013600  104100                                  HLT     ;AS             ;WHY DID AT BITS SET?
013602  112777  000100  165334                  MOVB    #100,@RSCS1B    ;CLEAR TRE
013610  032777  010000  165270                  BIT     #NED,@RSCS2     ;DID NED CLEAR
013616  001401                                  BEQ     .+4             ;YES
013620  104200                                  HLT     ;CS2            ;NED DID NOT CLEAR
013622  017767  165266  165350                  MOV     @RSDA,WORK      ;READ DRIVE REG
013630  032777  010000  165250                  BIT     #NED,@RSCS2     ;DID NED SET?
013636  001001                                  BNE     .+4
013640  104040                                  HLT     ;DS             ;NED DID NOT SET
013642  000431                                  BR      NNDD            ;GET OUT
013644  032767  010000  165312  NEDDON: BIT     #BIT12,ONCEE    ;DWAS THIS TYPED BEFORE?
013652  001025                                  BNE     NNDD            ;YES
013654  104402  013660                          TYPE    ,.+2            ;.ASCIZ <15><12>"COULD NOT FIND A NON-EXISTENT DRIVE"
013726  052767  010000  165230  NNDD:   BIS     #BIT12,ONCEE    ;SET TYPED FLAG
```

```
3664                                    ;*************************************************************************
3665                                    ;TEST 73                TEST THAT DAO IN RSER AND LBT IN RSDS DO SET
3666                                    ;*************************************************************************
3667  013734  104400                    TST73:  SCOPE
3668
3669  013736  104414                    DAOTST: CLRDK                           ;CLEAR ALL RS REG
3670  013740  012777  177577  165142            MOV     #-201,@RSWC             ;LOAD WC FOR RS04
3671  013746  005767  165214                    TST     RS04D?                  ;IS THIS A RS04?
3672  013752  001003                            BNE     1$                      ;YES
3673  013754  012777  177677  165126            MOV     #-101,@RSWC             ;NO
3674  013762  012777  007777  165124    1$:     MOV     #7777,@RSDA             ;SET RSDA=TO ALL ONES
3675  013770  013777  001102  165114    2$:     MOV     @#OBUFSV,@RSBA          ;CURRENT ADDRESS=OUTBUF
3676  013776  012777  000061  165100            MOV     #61,@RSCS1              ;WRITE
3677  014004  004767  010246                    JSR     PC,WAITRY               ;WAIT FOR READY
3678  014010  104001                            HLT     !CS1                    ;RDY DID NOT SET
3679  014012  022777  001000  165100            CMP     #1000,@RSER             ;DID DAO SET?
3680  014020  001401                            BEQ     .+4                     ;YES
3681  014022  104002                            HLT     !ER                     ;DAO DID NOT SET
3682  014024  022777  152600  165064            CMP     #152600,@RSDS           ;DID LBT SET?
3683  014032  001401                            BEQ     .+4                     ;YES
3684  014034  104040                            HLT     !DS                     ;LBT DID NOT SET
3685  014036  005777  165042                    TST     @RSCS1                  ;IS ERROR FLAG SET
3686  014042  100401                            BMI     .+4                     ;ERROR IS SET
3687  014044  104001                            HLT     !CS1                    ;SC DID NOT SET
3688  014046  104414                            CLRDK                           ;CLEAR ALL RS REG
3689  014050  022777  010600  165040            CMP     #10600,@RSDS            ;DID ATA +LBT CLEAR
3690  014056  001401                            BEQ     .+4                     ;YES
3691  014060  104040                            HLT     !DS                     ;ATA DID NOT CLEAR BY CLR BIT
3692  014062  005777  165032                    TST     @RSER                   ;DID DAO CLEAR?
3693  014066  001401                            BEQ     .+4                     ;YES
3694  014070  104002                            HLT     !ER                     ;DAO DID NOT CLEAR WITH CLEAR
```

```
3695                                    ;*************************************************************************
3696                                    ;TEST 74                TEST THAT LBT DOES SET AND DAO DOES NOT
3697                                    ;*************************************************************************
3698  014072  104400                    TST74:  SCOPE
3699
3700  014074  104414                    DAOTT:  CLRDK                           ;CLEAR ALL RS REG
3701  014076  012777  177601  165004            MOV     #-177,@RSWC             ;LOAD WC FOR RS04
3702  014104  005767  165056                    TST     RS04DT                  ;IS THIS A RS04?
3703  014110  001003                            BNE     1$                      ;YES
3704  014112  012777  177701  164770            MOV     #-77,@RSWC              ;NO
3705  014120  012777  007777  164766  1$:       MOV     #7777,@RSDA             ;SET RSDA=TO ALL ONES
3706  014126  013777  001102  164756  2$:       MOV     @#OBUFSV,@RSBA          ;CURRENT ADDRESS=OUTBUF
3707  014134  012777  000061  164742            MOV     #61,@RSCS1              ;WRITE
3708  014142  004767  010110                    JSR     PC,WAITRY               ;WAIT FOR READY
3709  014146  104001                            HLT     !CS1                    ;RDY DID NOT SET
3710  014150  005777  164744                    TST     @RSER                   ;ANY ERRORS?
3711  014154  001401                            BEQ     .+4                     ;NO
3712  014156  104002                            HLT     !ER                     ;YES
3713  014160  022777  012600  164730            CMP     #12600,@RSDS            ;DID LBT SET?
3714  014166  001401                            BEQ     .+4                     ;YES
3715  014170  104040                            HLT     !DS                     ;LBT DID NOT SET
3716  014172  005777  164706                    TST     @RSCS1                  ;IS ERROR FLAG SET
3717  014176  100001                            BPL     .+4                     ;NO
3718  014200  104001                            HLT     !CS1                    ;ERROR
3719  014202  104414                            CLRDK                           ;CLEAR ALL RS REG
3720  014204  022777  010600  164704            CMP     #10600,@RSDS            ;DID LBT CLEAR
3721  014212  001401                            BEQ     .+4                     ;YES
3722  014214  104040                            HLT     !DS                     ;ATA DID NOT CLEAR BY CLR BIT
```

```
                                ;*****************************************************************************
                                ;TEST 75                    EXECUTE FUNCTION WITH ERROR BITS SET
                                ;*****************************************************************************
014216  104400                  TST75:  SCOPE

                        ERTST:  CLRDK                           ;CLEAR ALL RS REG
014220  104414
014222  012777  177017  164670          MOV     #177017,@RSER   ;LOAD ER
014230  017700  164666                  MOV     @RSAS,BAD       ;GET AS REG
014234  016701  164722                  MOV     UNCMP,GOOD      ;GET UNIT ATA BIT
014240  042701  177400                  BIC     #177400,GOOD    ;CLEAR JUNK
014244  020100                          CMP     GOOD,BAD        ;IS AS REG CORRECT?
014246  001401                          BEQ     .+4             ;YES
014250  104100                          HLT     !AS             ;AS BIT SHOULD BE SET
014252  022777  104200  164624          CMP     #104200,@RSCS1  ;DID ERRS SET IN CS1?
014260  001401                          BEQ     .+4             ;YES
014262  104040                          HLT     !DS             ;CS1 SHOULD =104200
014264  016777  164672  164630          MOV     UNCMP,@RSAS     ;CLEAR ATA BIT
014272  005777  164624                  TST     @RSAS           ;DID IT CLEAR?
014276  001401                          BEQ     .+4             ;YES
014300  104100                          HLT     !AS             ;COULD NOT CLEAR AS BIT
                                                                ;BY LOADING A 1 INTO IT
014302  022777  004200  164574          CMP     #4200,@RSCS1    ;DID SC CLEAR BY
014310  001401                          BEQ     .+4             ;CLEARING ATA
014312  104002                          HLT     !ER             ;NO
014314  012767  177777  010226          MOV     #177777,OUTBUF  ;DATA TO BE XFERED
014322  013777  001102  164562          MOV     @#OBUFSV,@RSBA  ;SET UP CURRENT ADDRESS
014330  012777  177777  164552          MOV     #-1,@RSWC       ;LOAD WC WITH -1
014336  012777  000071  164540          MOV     #71,@RSCS1      ;DO READ FUNCTION
014344  032777  000001  164532          BIT     #1,@RSCS1       ;DID GO BIT CLEAR
014352  001401                          BEQ     .+4             ;YES
014354  104001                          HLT     !CS1            ;GO BIT SHOULD BE CLEARED
014356  105777  164522          1$:     TSTB    @RSCS1          ;WAIT FOR READY
014362  100375                          BPL     1$              ;WAIT
014364  022777  144270  164512          CMP     #144270,@RSCS1  ;DID ERRS CLEAR BY SETTING GO BIT?
014372  001401                          BEQ     .+4             ;YES
014374  104002                          HLT     !ER             ;NO
```

```
3759  014376  017700  164504                MOV     @RSCS2,BAD      ;GET CS2
3760  014402  012701  001100                MOV     #1100,GOOD      ;GET CORRECT ANS
3761  014406  056701  164544                BIS     UNNUM,GOOD      ;GET UNIT #
3762  014412  020100                        CMP     GOOD,BAD        ;IS CS2 CORRECT?
3763  014414  001401                        BEQ     .+4             ;YES
3764  014416  104000                        HLT                     ;GOOD = WHAT CS2 SHOULD =
3765  014420  022777  150600  164470        CMP     #150600,@RSDS   ;DID ERR BITS SET?
3766  014426  001401                        BEQ     .+4             ;NO
3767  014430  104040                        HLT     !DS             ;ERR BIT SHOULD BE 1
3768  014432  022777  177777  164450        CMP     #-1,@RSWC       ;DID WC MOVE?
3769  014440  001401                        BEQ     .+4             ;NO
3770  014442  104010                        HLT     !WC             ;WC SHOULD = 1777777
3771  014444  005777  164444                TST     @RSDA           ;DID DA MOV
3772  014450  001401                        BEQ     .+4             ;NO
3773  014452  104004                        HLT     !DA             ;DA SHOULD =0
3774  014454  023777  001102  164430        CMP     @#OBUFSV,@RSBA  ;DID BA MOVE
3775  014462  001401                        BEQ     .+4             ;NO
3776  014464  104020                        HLT     !BA             ;BA MOVED
3777  014466  036777  164470  164426        BIT     UNCMP,@RSAS     ;AS SHOULD BE SET
3778  014474  001001                        BNE     .+4             ;IS IT?
3779  014476  104100                        HLT     !AS             ;NO
3780  014500  022777  177017  164412        CMP     #177017,@RSER   ;DID ER CHANGE?
3781  014506  001401                        BEQ     .+4             ;NO
3782  014510  104002                        HLT     !ER             ;ER SHOULD NOT CHANGE
```

```
                                ;*****************************************************************************
                                ;TEST 76                    PAT AND MCPE TEST
                                ;*****************************************************************************
014512  104400                  TST76:  SCOPE

014514  104414                  PATST:  CLRDK                           ;CLEAR ALL RS REG
014516  052777  000020  164362          BIS     #BIT4,@RSCS2            ;SET PAT
014524  005777  164400                  TST     @RSMR                   ;READ DRIVE REG
014530  017700  164352                  MOV     @RSCS2,BAD              ;GET CS2
014534  012701  000120                  MOV     #120,GOOD               ;MDPE SHOULD
014540  056701  164412                  BIS     UNNUM,GOOD              ;NOT SET
014544  020100                          CMP     GOOD,BAD                ;IS CS2 CORRECT?
014546  001401                          BEQ     .+4                     ;YES
014550  104000                          HLT                             ;BAD = CS2 GOOD = CORRECT ANS
014552  012777  000010  164350          MOV     #10,@RSMR               ;CAUSE PAR TO SET IN RSER
014560  022777  000010  164332          CMP     #10,@RSER               ;DID PAR SET?
014566  001401                          BEQ     .+4                     ;YES
014570  104140                          HLT     !AS!DS
014572  017700  164324                  MOV     @RSAS,BAD               ;GET AS REG
014576  016701  164360                  MOV     UNCMP,GOOD              ;GET UNIT ATA BIT
014602  042701  177400                  BIC     #177400,GOOD            ;CLEAR JUNK
014606  020100                          CMP     GOOD,BAD                ;IS AS REG CORRECT?
014610  001401                          BEQ     .+4                     ;YES
014612  104100                          HLT     !AS                     ;AS BIT SHOULD BE SET
014614  022777  104200  164262          CMP     #104200,@RSCS1          ;DID ERRS SET IN CS1?
014622  001401                          BEQ     .+4                     ;YES
014624  104040                          HLT     !DS                     ;CS1 SHOULD =104200
014626  104414                          CLRDK                           ;CLEAR RS REG
014630  022777  004200  164246          CMP     #4200,@RSCS1            ;IS CS1 CORRECT?
014636  001401                          BEQ     .+4                     ;CLEARING ATA
014640  104002                          HLT     !ER                     ;NO
014642  017700  164240                  MOV     @RSCS2,BAD              ;CHECK TO SEE
014646  016701  164304                  MOV     UNNUM,GOOD              ;IF PAT CLEARS
014652  052701  000100                  BIS     #100,GOOD
014656  020100                          CMP     GOOD,BAD
014660  001401                          BEQ     .+4
014662  104000                          HLT                             ;PAT DID NOT CLEAR
```

```
                                ;*****************************************************************************
                                ;TEST 77                    SET PAT BIT AND LOAD FUNCTION
                                ;*****************************************************************************
014664  104400                  TST77:  SCOPE

014666  104414                  SETPAT: CLRDK                           ;CLEAR ALL REG
014670  052777  000010  164210          BIS     #BAI,@RSCS2             ;SET BAI
014676  012767  177777  007644          MOV     #177777,OUTBUF          ;DATA TO BE XFERED
014704  013777  001102  164200          MOV     @#OBUFSV,@RSBA          ;SET UP CURRENT ADDRESS
014712  012777  177000  164170          MOV     #-1000,@RSWC            ;LOAD WC WITH -1
014720  052777  000020  164160          BIS     #BIT4,@RSCS2            ;SET PAT BIT
014726  012777  000071  164150          MOV     #71,@RSCS1              ;DO READ FUNCTION
014734  105777  164144          1$:     TSTB    @RSCS1                  ;WAIT FOR READY
014740  100375                          BPL     1$                      ;WAIT
014742  022777  144270  164134          CMP     #144270,@RSCS1          ;DID CS1 GET LOADED?
014750  001401                          BEQ     .+4                     ;NO
014752  104040                          HLT     !DS                     ;IT SHOULD NOT
014754  023777  001102  164130          CMP     @#OBUFSV,@RSBA          ;DID BA MOVE?
014762  001401                          BEQ     .+4                     ;NO
014764  104020                          HLT     !BA                     ;YES
014766  022777  177000  164114          CMP     #-1000,@RSWC            ;DID WC MOVE?
014774  001401                          BEQ     .+4                     ;NO
014776  104010                          HLT     !WC                     ;YES WHY?

                                ;*****************************************************************************
                                ;TEST 100                   DO FUNCTION THEN SET PAT BIT
                                ;*****************************************************************************
015000  104400                  TST100: SCOPE
015002  104414                  FUNDO:  CLRDK                           ;CLEAR ALL REG
015004  052777  000010  164074          BIS     #BAI,@RSCS2             ;SET BAI
015012  012767  177777  007530          MOV     #177777,OUTBUF          ;DATA TO BE XFERED
015020  013777  001102  164064          MOV     @#OBUFSV,@RSBA          ;SET UP CURRENT ADDRESS
015026  012777  177000  164054          MOV     #-1000,@RSWC            ;LOAD WC WITH -1
015034  012777  000071  164042  3$:     MOV     #71,@RSCS1              ;DO A READ
015042  100774                          BMI     3$                      ;WAIT FOR BUSY
015044  052777  000020  164034          BIS     #BIT4,@RSCS2            ;SET PAT
015052  105777  164026          2$:     TSTB    @RSCS1                  ;WAIT FOR READY
015056  100375                          BPL     2$
015060  022777  144270  164016          CMP     #144270,@RSCS1          ;DID MCPE SET?
015066  001401                          BEQ     .+4                     ;NO
015070  104002                          HLT     !ER                     ;YES
015072  017700  164010                  MOV     @RSCS2,BAD              ;GET CS2
015076  012701  000730                  MOV     #730,GOOD               ;GET CORRECT ANS
015102  056701  164050                  BIS     UNNUM,GOOD              ;GET UNIT #
015106  020100                          CMP     GOOD,BAD                ;IS CS2 CORRECT?
015110  001401                          BEQ     .+4                     ;YES
015112  104000                          HLT                             ;GOOD = WHAT CS2 SHOULD =
```

```
3867  015114  022777  010600  163774          CMP     #10600,@RSDS    ;DID ERR BITS SET?
3868  015122  001401                          BEQ     .+4             ;NO
3869  015124  104040                          HLT     !DS             ;ERR BIT SHOULD BE 1
3870  015126  005777  163766                  TST     @RSER           ;IS ER CLEAR?
3871  015132  001401                          BEQ     .+4             ;YES
3872  015134  104044                          HLT     !DS!DA          ;NO ERRORS SHOULD BE SET
3873  015136  104414                          CLRDK                   ;CLEAR ALL RS REG
3874  015140  022777  004200  163736          CMP     #4200,@RSCS1    ;IS CS1 CORRECT?
3875  015146  001401                          BEQ     .+4             ;YES
3876  015150  104040                          HLT     !DS
3877                        ;*************************************************************************
3878                        ;TEST 101                TEST PAR BY SETTING PAT
3879                        ;*************************************************************************
3880  015152  104400        TST101: SCOPE
3881
3882  015154  104414        PATTST: CLRDK                   ;CLEAR ALL RS REG
3883  015156  052777  000010  163722          BIS     #BAI,@RSCS2     ;SET BAI
3884  015164  012767  177777  007356          MOV     #177777,OUTBUF  ;DATA TO BE XFERED
3885  015172  013777  001102  163712          MOV     @#OBUFSV,@RSBA  ;SET UP CURRENT ADDRESS
3886  015200  012777  177000  163702          MOV     #-1000,@RSWC    ;LOAD WC WITH -1
3887  015206  012777  000061  163670          MOV     #61,@RSCS1      ;DO WRITE FUNCTION
3888  015214  105777  163664          1$:     TSTB    @RSCS1          ;WAIT FOR READY
3889  015220  100775                          BMI     1$              ;WAIT
3890  015222  052777  000020  163656          BIS     #BIT4,@RSCS2    ;SET PAT
3891  015230  105777  163650          2$:     TSTB    @RSCS1          ;WAIT FOR READY
3892  015234  100375                          BPL     2$
3893  015236  022777  144260  163640          CMP     #144260,@RSCS1  ;DID MCPE SET?
3894  015244  001401                          BEQ     .+4             ;NO
3895  015246  104002                          HLT     !ER             ;YES
3896  015250  017700  163632                  MOV     @RSCS2,BAD      ;GET CS2
3897  015254  012701  000230                  MOV     #230,GOOD       ;GET CORRECT ANS-- DO NOT CHECK IR - REASON 50 CYCLE
3898  015260  056701  163672                  BIS     UNNUM,GOOD      ;GET UNIT #
3899  015264  042700  000100                  BIC     #BIT6,BAD       ;CLEAR IR BIT FOR CS2 COMPARE
3900  015270  020100                          CMP     GOOD,BAD        ;IS CS2 CORRECT?
3901  015272  001401                          BEQ     .+4             ;YES
3902  015274  104000                          HLT                     ;GOOD = WHAT CS2 SHOULD =
3903  015276  022777  150600  163612          CMP     #150600,@RSDS   ;DID ERR BITS SET?
3904  015304  001401                          BEQ     .+4             ;NO
3905  015306  104040                          HLT     !DS             ;ERR BIT SHOULD BE 1
3906  015310  022777  000010  163602          CMP     #10,@RSER       ;DID PAR SET?
3907  015316  001401                          BEQ     .+4             ;YES
3908  015320  104044                          HLT     !DS!DA          ;NO
3909  015322  104414                          CLRDK                   ;CLEAR ALL RS REG
3910  015324  022777  004200  163552          CMP     #4200,@RSCS1    ;IS CS1 CORRECT?
3911  015332  001401                          BEQ     .+4             ;YES
3912  015334  104040                          HLT     !DS
3913  015336  005777  163556                  TST     @RSER           ;DID PAR CLEAR?
3914  015342  001401                          BEQ     .+4             ;YES
3915  015344  104040                          HLT     !DS
```

```
                                ;*****************************************************************************
                                ;TEST 102                TEST THE ABILITY TO FILL THE LAST SECTOR
                                ;*****************************************************************************
015346  104400                  TST102: SCOPE

                        LASTSC: CLRDK                   ;CLEAR ALL RS REG
015350  104414
015352  012777  007777  163534        MOV     #7777,@RSDA     ;SET RSDA=TO ALL ONES
015360  012777  177700  163522        MOV     #-100,@RSWC     ;WORD COUNT=-100
015366  005767  163574                TST     RS04DT          ;IS THIS A RS04?
015372  001403                        BEQ     1$              ;NO
015374  012777  177600  163506        MOV     #-200,@RSWC     ;YES
015402  013777  001102  163502  1$:   MOV     @#OBUFSV,@RSBA  ;CURRENT ADDRESS=OUTBUF
015410  012777  000061  163466        MOV     #61,@RSCS1      ;WRITE
015416  004767  006634                JSR     PC,WAITRY       ;WAIT FOR READY
015422  104001                        HLT     !CS1            ;RDY DID NOT SET
015424  005777  163470                TST     @RSER           ;DID ANY ERROR BITS SET?
015430  001401                        BEQ     .+4             ;NO
015432  104002                        HLT     !ER             ;GOT AN ERROR
015434  022777  012600  163454        CMP     #12600,@RSDS    ;DID LBT SET?
015442  001401                        BEQ     .+4             ;YES
015444  104040                        HLT     !DS             ;LBT DID NOT SET
015446  005777  163432                TST     @RSCS1          ;IS ERROR FLAG SET
015452  100001                        BPL     .+4             ;ERROR IS SET
015454  104001                        HLT     !CS1            ;SC DID NOT SET
015456  104414                        CLRDK                   ;CLEAR ALL RS REG
015460  022777  010600  163430        CMP     #10600,@RSDS    ;DID ATA +LBT CLEAR
015466  001401                        BEQ     .+4             ;YES
015470  104040                        HLT     !DS             ;ATA DID NOT CLEAR BY CLR BIT
```

```
                                ;FILL SECTOR WITH ALL ONES.
                                ;NOW WRITE 1ST WORD IN SECTOR
                                ;TEST REMAINING 63 WORDS FOR 0
                                ;
                                ;*************************************************************************
                                ;TEST 103                    TEST FOR ZERO'S IN PARTIAL FILLED SECTOR
                                ;*************************************************************************
015472  104400                TST103: SCOPE
                                ;
015474  104414                SECT:   CLRDK                   ;CLEAR ALL RS REG
015476  012767  177777  007044        MOV     #-1,OUTBUF      ;PUT -1INTO OUTBUF
015504  013777  001102  163400        MOV     @#OBUFSV,@RSBA  ;SET UP CURRENT ADDR
015512  012777  177600  163370        MOV     #-200,@RSWC     ;LOAD WC FOR RS04
015520  005767  163442                TST     RS04DT          ;RS04?
015524  001003                        BNE     5$              ;YES
015526  012777  177700  163354        MOV     #-100,@RSWC     ;SET WORD COUNT TO -100
015534  052777  000010  163344  5$:   BIS     #BAI,@RSCS2     ;SET BAI BIT
015542  012777  000061  163334        MOV     #61,@RSCS1      ;WRITE
015550  105777  163330          3$:   TSTB    @RSCS1          ;IS RDY SET?
015554  100375                        BPL     3$              ;NO
015556  005077  163332                CLR     @RSDA           ;SET DSK ADDRESS TO 0
015562  012767  177777  006760        MOV     #-1,OUTBUF      ;PUT 177777 INTO OUTBUF
015570  013777  001102  163314        MOV     @#OBUFSV,@RSBA  ;SET UP CURRENT ADDR
015576  012777  177777  163304        MOV     #-1,@RSWC       ;SET WORD COUNT TO -1
015604  052777  000010  163274        BIS     #10,@RSCS2      ;SET BAI BIT
015612  012777  000061  163264        MOV     #61,@RSCS1      ;WRITE
015620  105777  163260          1$:   TSTB    @RSCS1          ;IS RDY SET?
015624  100375                        BPL     1$              ;NO
015626  042777  000010  163252        BIC     #10,@RSCS2      ;CLEAR BAI BIT
015634  005767  163326                TST     RS04DT          ;RS04?
015640  001404                        BEQ     7$              ;NO
015642  012767  000200  163330        MOV     #200,WORK       ;YES
015650  000403                        BR      8$              ;CONT
015652  012767  000100  163320  7$:   MOV     #100,WORK       ;SET UP BUFFER
015660  013701  001102          8$:   MOV     @#OBUFSV,R1     ;GET STARTING ADD OF BUF
015664  012721  177777                MOV     #-1,(R1)+       ;LOAD FIRST WD WITH -1
015670  005021                  6$:   CLR     (R1)+           ;LOAD REST WITH 0
015672  005367  163302                DEC     WORK            ;DONE YET?
015676  001374                        BNE     6$              ;NO
015700  005077  163210                CLR     @RSDA           ;SET DSK ADDRESS TO 0
015704  013777  001102  163200        MOV     @#OBUFSV,@RSBA  ;SET UP CURRENT ADDR
015712  005767  163250                TST     RS04DT          ;RS04?
015716  001404                        BEQ     9$              ;NO
015720  012777  177600  163162        MOV     #-200,@RSWC     ;YES
015726  000403                        BR      10$             ;CONT
015730  012777  177700  163152  9$:   MOV     #-100,@RSWC     ;SET WORD COUNT TO -100
015736  012777  000051  163140  10$:  MOV     #51,@RSCS1      ;WRITE CHECK
```

```
 3991  015744  032777  000200  163132  2$:     BIT     #200,@RSCS1     ;IS RDY SET?
 3992  015752  001774                          BEQ     2$              ;NO
 3993  015754  016701  163176                  MOV     UNNUM,GOOD      ;GET UNIT #
 3994  015760  052701  000100                  BIS     #100,GOOD       ;SET IR BIT
 3995  015764  017700  163116                  MOV     @RSCS2,BAD      ;GET CS2
 3996  015770  020100                          CMP     GOOD,BAD        ;IS CS2 CORRECT?
 3997  015772  001401                          BEQ     .+4             ;YES
 3998  015774  104002                          HLT     !ER             ;THERE WAS A WRITE CHECK ERROR
 3999
 4000                          ;*****************************************************************************
 4001                          ;TEST 104                IF MEMORY MANAGEMENT IS AVAILABLE CHECK THE EXTENDED MEMORY ADDR
 4002                          ;*****************************************************************************
 4003  015776  104400          TST104: SCOPE
 4004
 4005  016000  104414          EXTTST: CLRDK                           ;CLEAR ALL RS REG.
 4006  016002  016767  002460  163160          MOV     TIMES,TIMSV     ;SAVE LOOP #
 4007  016010  012767  000010  002450          MOV     #10,TIMES       ;LOOP 10 TIMES
 4008  016016  012767  016522  161760          MOV     #EXTTRP,4       ;SETUP TIMEOUT TRAP
 4009  016024  012767  000340  161754          MOV     #340,6
 4010  016032  005737  177572                  TST     @#SR0           ;IF MEMORY MANAGEMENT IS NOT
 4011                                                                  ;AVAILABLE THE PROGRAM WILL TRAP
 4012                                                                  ;AND TRANSFER TO END OF THE TEST
 4013  016036  012767  016514  161740          MOV     #EXTRP,4
 4014  016044  012737  007600  172356          MOV     #7600,@#KIPAR7  ;OPEN I/O REGISTERS
 4015  016052  005037  172340                  CLR     @#KIPAR0        ;FREE FIRST 4K
 4016  016056  012737  000200  172342          MOV     #200,@#KIPAR1   ;ENABLE SECOND 4K
 4017  016064  012737  002000  172344          MOV     #2000,@#KIPAR2
 4018  016072  012737  177406  172300          MOV     #400*256.-400+UP+RW,@#KIPDR0    ;SET KIPDR0=RW UP 400 BLOCKS
 4019  016100  012737  177406  172302          MOV     #400*256.-400+UP+RW,@#KIPDR1    ;SET KIPDR1=RW UP 400 BLOCKS
 4020  016106  012737  177406  172304          MOV     #400*256.-400+UP+RW,@#KIPDR2    ;SET KIPDR2=RW UP 400 BLOCKS
 4021  016114  012737  177406  172316          MOV     #400*256.-400+UP+RW,@#KIPDR7    ;SET KIPDR7=RW UP 400 BLOCKS
 4022  016122  012737  000001  177572          MOV     #1,@#SR0        ;TURN ON MEMORY MANAGEMENT
 4023  016130  012702  040000                  MOV     #40000,R2       ;R2 EQUALS BASE ADDR
```

```
4024  016134  012712  177777           7$:   MOV   #177777,(R2)    ;INSERT PATTERN INTO 200000
4025  016140  012777  177776  162742         MOV   #-2,@RSWC       ;SETUP WORDCOUNT
4026  016146  012777  177777  162736         MOV   #177777,@RSBA   ;SETUP BUS ADDR
4027  016154  012777  000061  162722         MOV   #61,@RSCS1      ;WRITE TWO WORDS ON DISK.  RSBA
4028                                                               ;STARTS AT 177777 TO FORCE CARRY
4029                                                               ;TO SET A16
4030  016162  105777  162716                 TSTB  @RSCS1          ;WAIT FOR READY
4031  016166  100375                         BPL   .-4
4032  016170  005777  162710                 TST   @RSCS1
4033  016174  100002                         BPL   1$
4034  016176  104046                         HLT   !ER!DA!DS       ;STATUS ERROR AFTER 2 WORD WRITE
4035  016200  000447                         BR    2$              ;USING MEX0
4036  016202  022777  004660  162674   1$:   CMP   #4660,@RSCS1     ;IS CS1 CORRECT
4037  016210  001402                         BEQ   3$              ;YES
4038  016212  104002                         HLT   !ER             ;CS2 DID NOT COMPARE
4039  016214  000441                         BR    2$
4040  016216  005012                   3$:   CLR   (R2)            ;CLEAR LOCATION 200000
4041  016220  005077  162670                 CLR   @RSDA           ;SETUP DA
4042  016224  012777  177777  162660         MOV   #177777,@RSBA   ;SETUP BA
4043  016232  012777  177776  162650         MOV   #-2,@RSWC       ;SETUP WC
4044  016240  012777  000071  162636         MOV   #71,@RSCS1      ;READ TWO WORDS INTO LOCATIONS
4045                                                               ;177777 AND 200000.
4046  016246  105777  162632                 TSTB  @RSCS1          ;WAIT FOR READY
4047  016252  100375                         BPL   .-4
4048  016254  005777  162624                 TST   @RSCS1          ;ANY ERRORS?
4049  016260  100002                         BPL   4$              ;BRANCH IF NO
4050  016262  104002                         HLT   !ER             ;ERROR OFTER READING 2 WORDS
4051  016264  000415                         BR    2$
4052  016266  022777  004670  162610   4$:   CMP   #4670,@RSCS1     ;IS CS1 CORRECT?
4053  016274  001402                         BEQ   5$              ;YES
4054  016276  104002                         HLT   !ER             ;CS1 DID NOT COMPARE
4055  016300  000407                         BR    2$              ;READ STARTING AT 177777
4056  016302  022712  177777           5$:   CMP   #177777,(R2)    ;WAS DATA READ INTO LOCATION
4057  016306  001404                         BEQ   2$              ;200000 CORRECTLY? - BRANCH IF YES
4058  016310  012701  177777                 MOV   #177777,GOOD
4059  016314  011200                         MOV   (R2),BAD
4060  016316  104000                         HLT                   ;DATA COMPARE ERROR AT 200000
4061  016320  000240                   2$:   NOP
```

```
4062  016322  104414                    EXTT1:  CLRDK                   ;CLEAR ALL REG
4063  016324  012737  004000  172344            MOV     #4000,@#KIPAR2
4064  016332  012702  040000                    MOV     #40000,R2       ;R2 EQUALS THE BASE ADDR
4065  016336  012712  177777              7$:   MOV     #177777,(R2)    ;INSERT PATTERN INTO 400000
4066  016342  012777  177777  162542            MOV     #177777,@RSBA   ;SETUP BUS ADDR
4067  016350  012777  177776  162532            MOV     #-2,@RSWC       ;LOAD WC
4068  016356  012777  000461  162520            MOV     #461,@RSCS1     ;SET BIT A16 AND WRITE
4069  016364  105777  162514                    TSTB    @RSCS1          ;WAIT FOR READY
4070  016370  100375                            BPL     .-4
4071  016372  005777  162506                    TST     @RSCS1          ;ANY ERRORS?
4072  016376  100001                            BPL     4$              ;BRANCH IF NO
4073  016400  104002                            HLT     !ER             ;ERROR AFTER READING 2 WORDS
4074  016402  022777  005260  162474    4$:     CMP     #5260,@RSCS1    ;IS CS1 CORRECT?
4075  016410  001401                            BEQ     10$             ;BRANCH IF YES
4076  016412  104002                            HLT     !ER             ;NO CS1 DID NOT COMPARE
4077                                                                    ;READ STARTING AT 377777
4078  016414  005012                    10$:    CLR     (R2)            ;CLEAR LOCATION 400000
4079                                                                    ;READ TWO WORDS STARTING AT 377777
4080  016416  012777  177776  162464            MOV     #-2,@RSWC       ;SETUP WC
4081  016424  005077  162464                    CLR     @RSDA           ;SETUP DA
4082  016430  012777  177777  162454            MOV     #177777,@RSBA
4083  016436  012777  000471  162440            MOV     #471,@RSCS1     ;CLEAR A 17 SET A16,READ
4084  016444  105777  162434                    TSTB    @RSCS1          ;WAIT FOR READY
4085  016450  100375                            BPL     .-4
4086  016452  005777  162426                    TST     @RSCS1          ;ANY ERRORS?
4087  016456  100002                            BPL     11$             ;BRANCH IF NO
4088  016460  104002                            HLT     !ER             ;ERROR WHILE READING TWO WORDS
4089  016462  000414                            BR      EXTRP
4090  016464  022777  005270  162412    11$:    CMP     #5270,@RSCS1    ;IS CS1 CORRECT?
4091  016472  001401                            BEQ     12$             ;BRANCH IF YES
4092  016474  104002                            HLT     !ER             ;CS1 DID NOT COMPARE
4093                                                                    ;READ STARTING AT 377777
4094  016476  022712  177777            12$:    CMP     #177777,(R2)    ;WAS DATA READ INTO LOCATION 400000
4095  016502  001404                            BEQ     EXTRP           ;CORRECTLY? - BRANCH IF YES
4096  016504  012701  177777                    MOV     #177777,GOOD
4097  016510  011200                            MOV     (R2),BAD
4098  016512  104000                            HLT
4099                                                                    ;DATA COMPARE ERROR AT 400000 IF
                                                                        ;RECEIVED=0 - LOCATION WASN'T ACCESSED
4100  016514  005037  177572           EXTRP:  CLR     @#SRD           ;TURN OFF MEMORY MANAGEMENT
4101  016520  000401                            BR      EXT1
4102  016522  000240                    EXTTRP: NOP                     ;UPDATE TEST NUMBERS
4103  016524  012706  000500            EXT1:   MOV     #500,SP         ;RESTORE STACK
4104  016530  012767  000006  161246    MEMOUT: MOV     #6,4
4105  016536  005067  161244                    CLR     6
```

```
                                ;*****************************************************************************
                                ;TEST 105                TEST PROGRAM INTERRUPT BY MOVING 300 INTO RSCS1
                                ;*****************************************************************************
016542  104400                  TST105: SCOPE

016544  104414                  QES:    CLRDK                           ;CLEAR ALL DRIVES
016546  012706  000500                  MOV     #500,SP                 ;SETUP STACK
016552  012777  016624  162360          MOV     #PGTRAP,@RSVEC          ;SET UP VECTOR
016560  012777  000340  162354          MOV     #340,@RSVCPS            ;SET TRAP PS
016566  012737  000200  177776          MOV     #200,@#PS               ;SET PS AT PRIORITY 4
016574  012777  000300  162302          MOV     #300,@RSCS1             ;THIS SHOULD CAUSE A TRAP
016602  012767  000500  162370          MOV     #500,WORK               ;SETUP LOOP
016610  005367  162364                  1$:     DEC     WORK            ;DEC LOOP SHOULD
016614  001375                          BNE     1$                      ;INTERRUPE BEFORE LOOP IS DONE
016616  104001                          HLT     !CS1                    ;SHOULD NEVER GET HERE
016620  000167  000014                  JMP     QESDON                  ;GET OUT

016624  022626                  PGTRAP: CMP     (6)+,(6)+               ;TRAP OK
016626  022777  004200  162250          CMP     #4200,@RSCS1            ;DID IE CLEAR?
016634  001401                          BEQ     .+4                     ;YES
016636  104001                          HLT     !CS1                    ;IE SHOULD BE CLEARED
016640                          QESDON:

                                ;*****************************************************************************
                                ;TEST 106                TEST THAT DISK DOES NOT INTERRUPT WHEN PS IS AT 5
                                ;*****************************************************************************
016640  104400                  TST106: SCOPE

016642  012706  000500          INTR5:  MOV     #500,SP                 ;SETUP STACK
016646  016767  162316  001612          MOV     TIMSV,TIMES             ;RESTORE LOOP COUNTER
016654  104414                          CLRDK                           ;CLEAR ALL RS REG
016656  012777  016736  162254          MOV     #INT112,@RSVEC          ;SET UP INTERRUPT VECTOR
016664  012777  000340  162250          MOV     #340,@RSVCPS            ;SET PRIO.
016672  012737  000240  177776          MOV     #240,@#PS               ;LOCK OUT ALL INTERRUPTS ABOVE
016700  013700  177776                  MOV     @#PS,BAD                ;GET PS
016704  012777  177777  162176          MOV     #177777,@RSWC           ;SET WORD COUNT TO -1
016712  013777  001102  162172          MOV     @#OBUFSV,@RSBA          ;LOAD CURRENT ADDRESS
016720  012777  000161  162156          MOV     #161,@RSCS1             ;GO WRITE (INTERRUPT ENABLED)
016726  004767  005324                  JSR     PC,WAITRY               ;WAIT FOR READY
016732  104001                          HLT     !CS1                    ;NO RDY NEVER CAME UP
016734  000403                          BR      INTDON                  ;RESTART ROUTINE
                                ;PROCESSOR SHOULD NOT TRAP TO INT112

016736  012701  000240          INT112: MOV     #240,GOOD               ;WHAT PS SHOULD HAVE
016742  104000                          HLT                             ;GOOD = CORRECT ANS FOR PS
016744                          INTDON:                                 ;DONE GET OUT
```

```
                                ;*****************************************************************************
                                ;TEST 107                   TEST THAT DISK DOES INTERRUPT WHEN PS IS AT 4
                                ;*****************************************************************************
016744  104400                  TST107: SCOPE

016746  012706  000500          INTR4:  MOV     #500,SP         ;SETUP STACK
016752  104414                          CLRDK                   ;CLEAR ALL RS REG
016754  012777  017050  162156          MOV     #INT114,@RSVEC  ;SET UP DISK TRAP VECTOR
016762  012777  000340  162152          MOV     #340,@RSVCPS    ;SET PRIO.
016770  012737  000200  177776          MOV     #200,@#PS       ;SET PROCESSOR TO PRIORITY 4
016776  013700  177776                  MOV     @#PS,BAD        ;GET PS
017002  012701  000200                  MOV     #200,GOOD       ;GET CORRECT PS
017006  012777  177777  162074          MOV     #177777,@RSWC   ;SET WORD COUNT TO -1
017014  013777  001102  162070          MOV     @#OBUFSV,@RSBA  ;LOAD CURRENT ADDRESS
017022  012777  000161  162054          MOV     #161,@RSCS1     ;WRITE (INTERRUPT ENABLE
017030  005067  162144                  CLR     WORK
017034  005267  162140                  INC     WORK            ;WAIT FOR INTERRUPT TO OCCUR
017040  001375                          BNE     .-4
017042  104000                          HLT                     ;GOOD=CORRECT PS BAD=WRONG PS
017044  104042                          HLT     !ER!DS
017046  000405                          BR      DONINT          ;CONT
017050  022777  004260  162026  INT114: CMP     #4260,@RSCS1    ;DID IE CLEAR?
017056  001401                          BEQ     .+4             ;YES
017060  104001                          HLT     !CS1            ;WHY DID NOT IE CLEAR
017062                          DONINT:

                                ;*****************************************************************************
                                ;TEST 110                   TEST INTERRUPT ON ERROR
                                ;*****************************************************************************
017062  104400                  TST110: SCOPE

017064  012706  000500          ERINT:  MOV     #500,SP         ;SETUP STACK
017070  012737  000200  177776          MOV     #200,@#PS       ;SET PS AT PRI 4
017076  012777  017154  162034          MOV     #ERRINT,@RSVEC  ;SET UPINTERRUPT ADD.
017104  104414                          CLRDK                   ;CLEAR ALL RS REG
017106  012777  000340  162026          MOV     #340,@RSVCPS    ;SET PRIO.
017114  012777  177777  161772          MOV     #177777,@RSDA   ;SET RSDA=TO ALL ONES
017122  012777  177600  161760          MOV     #177600,@RSWC   ;WORD COUNT=-200
017130  013777  001102  161754          MOV     @#OBUFSV,@RSBA  ;CURRENT ADDRESS=OUTBUF
017136  012777  000161  161740          MOV     #161,@RSCS1     ;WRITE
017144  004767  005106                  JSR     PC,WAITRY       ;WAIT FOR READY
017150  104042                  1$:     HLT     !ER!DS          ;Y DIDN'T PGM INTERRUPT IS RDY SET?
017152  000406                          BR      FINTST          ;GET OUT
017154  022777  144260  161722  ERRINT: CMP     #144260,@RSCS1  ;IS CS1 RIGHT?
017162  001401                          BEQ     .+4             ;YES
017164  104042                          HLT     !ER!DS
017166  022626                          CMP     (6)+,(6)+       ;CLEAR STACK
017170                          FINTST:
```

```
                                ;*****************************************************************************
                                ;TEST 111                    DYNAMIC FUNCTION TEST
                                ;*****************************************************************************
017170  104400                  TST111: SCOPE
                                ;EXECUTE FUNCTION MODIFY UNIT # AND DO A DRIVE SEARCH
                                ;DRIVE SEARCH WILL ONLY BE DONE IF THERE ARE AT LEAST 2 DRIVES
                                ;2ND DRIVE MAY NOT BE TESTED YET SO IF THIS TEST FAILS CHECK 2ND DRIVE
                                ;BEFORE TRYING TO DEBUG THIS TEST

017172  104414                  MODNUM: CLRDK                   ;CLEAR ALL RS REG
017174  016767  001266  161766          MOV     TIMES,TIMSV     ;SAVE LOOP COUNT
017202  012767  000010  001256          MOV     #10,TIMES       ;LOOP ONLY 10 TIMES
017210  005067  161766                  CLR     WORK1           ;CLEAR WORK LOC.
017214  005003                          CLR     R3
017216  005004                          CLR     R4
017220  012702  020120                  MOV     #DVTAB,R2       ;SETUP TABLE
017224  012767  000401  161746          MOV     #401,WORK       ;SETUP TO TEST FOR MORE DRIVES
017232  036767  161742  161720  7$:     BIT     WORK,UNITSV     ;IS DRIVE ON SYSTEM?
017240  001403                          BEQ     6$              ;NO
017242  020467  161710                  CMP     R4,UNNUM        ;IS THIS THE SAME DRIVE?
017246  001017                          BNE     8$              ;NO
017250  005204                  6$:     INC     R4              ;UPDATE 2ND UNIT #
017252  000241                          CLC
017254  006167  161720                  ROL     WORK            ;CHECK FOR NEXT DRIVE
017260  103364                          BCC     7$              ;NOT DONE YET
017262  032767  000010  161712          BIT     #BIT3,WORK1     ;MULTI DRIVE?
017270  001016                          BNE     12$             ;YES
017272  016705  161660                  MOV     UNNUM,R5        ;LOAD UNIT NO
017276  005205                          INC     R5              ;CHANGE IT
017300  042705  177770                  BIC     #177770,R5      ;CLEAR JUNK
017304  000410                          BR      12$
017306  052767  000010  161666  8$:     BIS     #BIT3,WORK1     ;SET FOUND MULTI DRIVE
017314  010422                          MOV     R4,(R2)+        ;LOAD UNT # INTO TABLE
017316  005203                          INC     R3              ;COUNT # OF DRIVES
017320  010367  161652                  MOV     R3,SAVEE        ;SAVE IT
017324  000751                          BR      6$
017326  012777  017442  161604  12$:    MOV     #TSTVEC,@RSVEC  ;SETUP INT. TRAP
017334  012777  000340  161600          MOV     #340,@RSVCPS    ;SETUP PRIO.
017342  005067  005202                  CLR     OUTBUF          ;CLR TO READ INTO
017346  013777  001102  161536          MOV     @#OBUFSV,@RSBA  ;SET UP CURRENT ADDRESS
017354  012777  177000  161526          MOV     #-1000,@RSWC    ;SET WORD COUNT
017362  012777  000060  161524  15:     MOV     #60,@RSDA       ;LOAD DA
017370  012702  020120                  MOV     #DVTAB,R2       ;GET TABLE
017374  012777  000161  161502          MOV     #161,@RSCS1     ;GO WRITE
017402  005703                          TST     R3              ;MORE THEN 1 DRIVE?
017404  001003                          BNE     13$             ;YES
017406  010577  161474                  MOV     R5,@RSCS2       ;NO MODIFY UNIT #
017412  000407                          BR      14$
```

```
4248  017414  012277  161466          13$:    MOV     (R2)+,@RSCS2    ;LOAD UNIT#
4249  017420  012777  000131  161456          MOV     #131,@RSCS1     ;DO SEARCH
4250  017426  005303                          DEC     R3              ;DONE ALL DRIVES YET?
4251  017430  001371                          BNE     13$             ;NO
4252  017432  012737  000200  177776  14$:    MOV     #200,@#PS       ;ENABLE INTERRUPTS
4253  017440  000001                  WTDV:   WAIT
4254  017442  016777  161510  161436  TSTVEC: MOV     UNNUM,@RSCS2    ;GET 1ST DRIVE
4255  017450  017700  161430                  MOV     @RSCS1,BAD      ;GET CS1
4256  017454  042700  100000                  BIC     #BIT15,BAD      ;CLEAR SC
4257  017460  012701  004260                  MOV     #4260,GOOD      ;GET CORRECT ANS
4258  017464  020100                          CMP     GOOD,BAD        ;IS CS1 CORRECT?
4259  017466  001402                          BEQ     4$              ;NO! X-FER OK
4260  017470  104000                          HLT                     ;CS1 SHOULD = 14270 OR 4270
4261  017472  104140                          HLT     !DS!AS
4262  017474  005777  161410          4$:     TST     @RSWC           ;TEST WC
4263  017500  001401                          BEQ     .+4             ;WORD COUNT DID OVERFLOW
4264  017502  104010                          HLT     !WC             ;SHOULD = 0
4265  017504  016701  161446                  MOV     UNNUM,GOOD      ;GET CORRECT
4266  017510  052701  000100                  BIS     #100,GOOD       ;ANS OF CS2
4267  017514  017700  161366                  MOV     @RSCS2,BAD      ;GET CS2
4268  017520  020100                          CMP     GOOD,BAD        ;IS CS2 CORRECT?
4269  017522  001401                          BEQ     .+4             ;YES
4270  017524  104000                          HLT                     ;GOOD = CORRECT ANS FOR CS2
4271  017526  017700  161360                  MOV     @RSBA,BAD       ;FETCH CURRENT ADDRESS
4272  017532  013701  001102                  MOV     @#OBUFSV,GOOD   ;WHAT RSBA SHOULD EQUAL
4273  017536  062701  002000                  ADD     #2000,GOOD      ;UPDATE IT
4274  017542  020001                          CMP     BAD,GOOD        ;IS RSBA CORRECT
4275  017544  001401                          BEQ     .+4             ;YES EXECUTE CONTINUE
4276  017546  104000                          HLT                     ;RSBA FAILED TO INCREMENT
4277  017550  005767  161412          2$:     TST     RS04DT          ;IS THIS A RS04?
4278  017554  001005                          BNE     5$              ;YES
4279  017556  022777  000070  161330          CMP     #70,@RSDA       ;IS DA CORRECT?
4280  017564  001406                          BEQ     3$              ;YES
4281  017566  104104                          HLT     !DA!AS          ;DA NOT CORRECT
4282  017570  022777  000064  161316  5$:     CMP     #64,@RSDA       ;WAS RSDA INCREMENTED
4283  017576  001401                          BEQ     .+4             ;RSDA OK
4284  017600  104046                          HLT     !DA!ER!DS       ;RSDA SHOULD CONTAIN A 64
4285  017602  012777  040000  161274  3$:     MOV     #TRE,@RSCS1     ;CLEAR ALL ERRORS IF ANY
4286  017610  032767  000010  161364          BIT     #BIT3,WORK1     ;MULTI DRIVE?
4287  017616  001461                          BEQ     WTDV1           ;NO
4288  017620  012777  017644  161312  1$:     MOV     #TTVEC,@RSVEC   ;SETUP INT FOR NEXT DRIVE
4289  017626  012716  017642                  MOV     #WTDV2,(SP)     ;GET WAIT
4290  017632  012777  000100  161244          MOV     #100,@RSCS1     ;SET IE
4291  017640  000002                          RTI                     ;RETURN
4292  017642  000001                  WTDV2:  WAIT
```

```
                                        ;SERVICE ROUTINE FOR SEARCH FUNCTIONS
017644 005002                   TTVEC:  CLR     R2              ;CLEAR UNIT #
017646 000241                           CLC
017650 012767 000401 161322             MOV     #401,WORK
017656 036777 161316 161236     1$:     BIT     WORK,@RSAS      ;DID THIS DRIVE INT?
017664 001006                           BNE     2$              ;YES
017666 005202                           INC     R2              ;UPDATE UNIT #
017670 000241                           CLC
017672 006167 161302                    ROL     WORK
017676 103367                           BCC     1$
017700 104100                           HLT     !AS             ;WHY DID WE INT WITH NO ATA???
017702 010277 161200            2$:     MOV     R2,@RSCS2       ;GET DRIVE
017706 022777 110600 161202             CMP     #110600,@RSDS   ;DID PIP CLEAR?
017714 001401                           BEQ     .+4             ;YES
017716 104140                           HLT     !DS!AS          ;PIP BIT DID NOT CLEAR
017720 022777 104230 161156             CMP     #104230,@RSCS1  ;DID SC SET?
017726 001401                           BEQ     .+4             ;YES
017730 104140                           HLT     !AS!DS          ;SC DID NOT SET
017732 005367 161240                    DEC     SAVEE           ;COUNT # OF INT
017736 001411                           BEQ     WTDV1           ;DONE YET?
017740 016777 161234 161154             MOV     WORK,@RSAS      ;CLEAR AS
017746 012777 000100 161130             MOV     #100,@RSCS1     ;SET IE
017754 012716 017642                    MOV     #WTDV2,(SP)     ;RETURN TO WAIT
017760 000002                           RTI
017762 012737 000340 177776     WTDV1:  MOV     #340,@#PS
017770 012706 000500                    MOV     #500,SP         ;CLEAR STACK
017774 016777 161142 161136             MOV     RSVCPS,@RSVEC   ;RESTORE INT VECTOR
020002 005077 161134                    CLR     @RSVCPS
020006 104400                   MODDON: SCOPE                   ;DONE
020010 016767 161154 000450             MOV     TIMSV,TIMES     ;RESTORE LOOP COUNT
020016 012737 000340 177776             MOV     #340,@#PS       ;RESTORE PS
020024 012767 000001 160746             MOV     #1,ICNT         ;FUGE TEST NUMBERS
020032 000137 001606            OUT:    JMP     @#TRYNX         ;TEST NEXT DRIVE
                                .SBTTL          $DONE - BELL AND SCOPE ROUTINE

020036 104400                   DONE:   SCOPE                   ;TERMINATIONG SCOPE FOR LOOPING
020040 062767 000001 160740             ADD     #1,PCNT+2       ;ADD 1 TO THE PASS COUNT
020046 005567 160732                    ADC     PCNT            ;MAKE IT DOUBLE PREC.
020052 032737 002000 177570             BIT     #SW10,@#SWR     ;RING THE BELL?
020060 001004                           BNE     4$              ;NO!
020062 104402 020066                    TYPE    .+2             ;.ASCIZ <BELL><177>
020072 013700 000042            4$:     MOV     @#42,R0         ;GET MONITOR ADDRESS
020076 001405                           BEQ     SEND1           ;IF NONE
020100 000005                           RESET
020102 004710                   SENDAD: JSR     7,(0)           ;GO TO MONITOR
020104 000240 000240 000240             240,240,240             ;SAVE ROOM FOR ACT11
020112 000137 001206            SEND1:  JMP     @#BEGIN         ;RETURN

020116 000000                   .TBIT:  0                       ;T BIT FLAG

020120 000010                   DVTAB:  .BLKW   10
```

```
                                .SBTTL          $TYPE - TTY TYPEOUT ROUTINE

                                ;THIS ROUTINE IS USE TO TYPE ASCII MESSAGES ON THE TTY. THE
                                ;CALL CAN BE IN ONE OF 3 FORMS: 1) "TYPE ,ADR" - TYPES THE
                                ;MESSAGE STARTING IN LOCATION "ADR:", 2) "TYPE ,CHAR" - TYPES
                                ;THE ASCII "CHAR", AND 3) "PRINT <<15><12>"MESSAGE"> - TYPES
                                ;THE MESSAGE WHICH IS INLINE ASCII. THE FILLER CHARACTER WHICH IS
                                ;TYPED AFTER A LINE FEED IS IN FILCHR AND THE NUMBER OF FILLERS
                                ;IS IN FILCHR+1.

020140  010446                .TYPE:  MOV     R4,-(6)         ;SAVE R4
020142  010546                        MOV     R5,-(6)         ;SAVE R5
020144  017605  000004                MOV     @4(6),R5        ;GET ADDRESS TO BE TYPED
020150  032705  177400                BIT     #177400,R5      ;IS IT A TYPEM?
020154  001002                        BNE     1$              ;NO
020156  016605  000004                MOV     4(6),R5         ;GET ADDRESS OF CHARACTER
020162  105715                1$:     TSTB    (R5)            ;TERMINATOR?
020164  001423                        BEQ     2$              ;GET OUT IF SO
020166  122715  000012                CMPB    #12,(R5)        ;IS THE CHAR A LINE FEED
020172  001012                        BNE     4$              ;NO - GET OUT
020174  116704  160615                MOVB    FILCHR+1,R4     ;GET THE FILL COUNT
020200  116777  160610  160612  5$:   MOVB    FILCHR,@TPB     ;TYPE A FILLER
020206  105777  160604                TSTB    @TPS            ;DONE YET?
020212  100375                        BPL     .-4             ;NO - WAIT
020214  005304                        DEC     R4              ;DEC COUNT
020216  001370                        BNE     5$              ;LOOP UNTIL 0
020220  112577  160574        4$:     MOVB    (R5)+,@TPB      ;LOAD AND TYPE THE CHARACTER
020224  105777  160566                TSTB    @TPS            ;IS THE PRINTER READY
020230  100375                        BPL     .-4             ;WAIT UNTIL IT IS
020232  000753                        BR      1$              ;GET THE NEXT CHARACTER
020234  017646  000004        2$:     MOV     @4(6),-(6)      ;GET ADDRESS TO BE TYPED
020240  062766  000002  000006        ADD     #2,6(6)         ;ADD 2 TO THE ADDRESS
020246  022666  000004                CMP     (6)+,4(6)       ;IS IT .+2?
020252  001006                        BNE     3$              ;NO
020254  062705  000002                ADD     #2,R5           ;ADD 2 TO THE ADDRESS
020260  042705  000001                BIC     #1,R5           ;BACK UP TO AN EVEN BYTE
020264  010566  000004                MOV     R5,4(6)         ;RESTORE ADDRESS
020270  012605                3$:     MOV     (6)+,R5         ;RESTORE R5
020272  012604                        MOV     (6)+,R4         ;RESTORE R4
020274  000002                        RTI                     ;RETURN
```

```
                                .SBTTL          $SCOPE - SCOPE LOOP HANDLER

                                ;THIS ROUTINE HANDLES THE ITERATIONS, LOOPING, ERROR
                                ;LOOPING, AND THE DISPLAYING OF THE TEST NUMBER.
                                ;"SCOPE" IS PLACED BETWEEN EACH SUBTEST IN THE TEST AND
                                ;RECORDS THE STARTING ADDRESS OF THE SUBTEST IN "LAD:"

020276  032737  000400  177570  .SCOPE: BIT     #SW8,@#SWR      ;LOOP ON SPEC. TEST?
020304  001404                          BEQ     1$              ;NO LOOP ON SPEC. TEST
020306  123767  177570  160464          CMPB    @#SWR,ICNT      ;ON RIGHT TEST?   *SW7-0*
020314  001453                          BEQ     .OVER           ;NOT RIGHT TEST
020316  032737  040000  177570  1$:     BIT     #SW14,@#SWR     ;LOOP ON TEST?
020324  001045                          BNE     .KIT            ;LOOP ON TEST IS SET
020326  000416                          BR      3$              ;SKIP - NOP FOR XOR TESTER
020330  013746  000004                  MOV     @#4,-(6)        ;PUSH @#4 ON STACK
020334  012737  020354  000004          MOV     #4$,@#4         ;SET FOR TIMEOUT
020342  005737  177060                  TST     @#177060        ;ERROR ON XOR?
020346  012637  000004                  MOV     (6)+,@#4        ;POP STACK INTO @#4
020352  000422                          BR      .SVLAD          ;NO ERROR - GO TO NEXT TEST
020354  022626                  4$:     CMP     (6)+,(6)+       ;CLEAR STACK
020356  012637  000004                  MOV     (6)+,@#4        ;POP STACK INTO @#4
020362  000426                          BR      .KIT            ;ERROR - LOOP ON TEST
020364  032737  004000  177570  3$:     BIT     #SW11,@#SWR     ;KILL ITERATIONS
020372  001012                          BNE     .SVLAD          ;YES - KILL ITERATIONS
020374  105767  160401                  TSTB    ICNT+1          ;FIRST ONE?
020400  001404                          BEQ     2$              ;BRANCH IF FIRST
020402  126767  000060  160371          CMPB    TIMES,ICNT+1    ;DONE?
020410  003013                          BGT     .KIT            ;BRANCH IF NOT
020412  112767  000001  160361  2$:     MOVB    #1,ICNT+1       ;FIRST ITERATION
020420  105267  160354          .SVLAD: INCB    ICNT            ;COUNT TEST NUMBERS
020424  011667  160360                  MOV     (6),LAD         ;SAVE LOOP ADDRESS
020430  016737  160344  177570          MOV     ICNT,@#DISPLAY  ;DISPLAY TEST NO. AND ITERATION COUNT
020436  000002                          RTI                     ;RETURN

020440  105267  160335          .KIT:   INCB    ICNT+1          ;INC THE ITERATION COUNT
020444  016737  160330  177570  .OVER:  MOV     ICNT,@#DISPLAY  ;SET UP DISPLAY
020452  005767  160332                  TST     LAD             ;FIRST ONE?
020456  001760                          BEQ     .SVLAD          ;YES
020460  016716  160324                  MOV     LAD,(6)         ;FUDGE RETURN ADDRESS
020464  000002                          RTI                     ;FIXES PS

020466  000100                  TIMES:  100                     ;RUN 100 TIMES
```

```
                                .SBTTL          $HLT - HLT ROUTINE (ERROR TYPEOUT)

                                ;THIS ROUTINE PRINTS OUT ERROR MESSAGES STARTING WITH THE
                                ;ADDRESS OF THE "HLT". IT ALSO COUNTS THE NUMBER OF ERRORS
                                ;AND HAS THE CAPABILITY OF LOOPING ON ERROR, BELL ON ERROR,
                                ;"HALT" ON ERROR, AND INHIBIT TYPEOUTS. AN OPTIONAL ARGUMENT
                                ;(HLT+3) WILL BE PLACED IN ".HLTCT:" FOR ADITIONAL TYPEOUTS.

020470  032737  002000  177570  .HLT:   BIT     #SW10,@#SWR     ;BELL ON ERROR?
020476  001402                          BEQ     1$              ;NO - SKIP
020500  104402  000007                  TYPE    ,BELL           ;RING BELL
020504  005267  160272          1$:     INC     ERRORS          ;COUNT THE NUMBER OF ERRORS
020510  032737  020000  177570          BIT     #SW13,@#SWR     ;SKIP TYPEOUT IF SET
020516  001025                          BNE     2$              ;SKIP TYPEOUTS
020520  104402  020524                  TYPE    ,.+2            ;.ASCIZ <15><12>
020530  011667  160256                  MOV     (6),HLTADR      ;PUT ADDRESS OF INSTRUCTION ON STACK
020534  162767  000002  160250          SUB     #2,HLTADR       ;FUDGE ADDRESS
020542  117767  160244  000054          MOVB    @HLTADR,.HLTCT  ;GET HLT ARGUEMENT
020550  016746  160236                  MOV     HLTADR,-(6)     ;PUT HLTADR ON STACK
020554  104404                          TYPEO                   ;TYPE STACK IN OCTAL
020556  104402  020562                  TYPE    ,.+2            ;.ASCIZ "  "
020566  004767  002752                  JSR     PC,RSREG        ;GO TO USER ERROR ROUTINE
020572  005737  177570          2$:     TST     @#SWR           ;HALT ON ERROR
020576  100001                          BPL     .+4             ;SKIP IF CONTINUE
020600  000000                          HALT                    ;HALT ON ERROR!
020602  032737  001000  177570          BIT     #SW9,@#SWR      ;CHECK FOR INHIBIT LOOP ON ERROR
020610  001003                          BNE     3$              ;SKIP IF LOOP ON ERROR
020612  105067  160163                  CLRB    ICNT+1          ;CLEAR ITERATION COUNT
020616  000002                          RTI                     ;RETURN
020620  000167  177614          3$:     JMP     .KIT            ;LOOP ON TEST UNTIL NO ERRORS

020624  000000                  .HLTCT: 0                       ;HLT ARGUMENT
```

```
                                        .SBTTL          $OCTAL - OCTAL TYPEOUT ROUTINE

                                        ;THIS ROUTINE IS USED TO TYPE AN OCTAL NUMBER ON THE TTY. IT WILL TYPE
                                        ;ALL 6 CHARACTERS, SUPPRESS LEADING ZEROES, OR TYPE THE
                                        ;16 BITS. IT IS CALLED VIA THE TYPOCT, TYPBIT, OR TYPOCS MACRO'S.

020626  012767  170101  000160  .TYPEB: MOV     #170101,.PR     ;SET BIT FLAG AND 16. CHARACTER COUNT
020634  000411                  BR      .PTIT           ;NOW TYPE IT IN BIT FORM
020636  112767  000001  000150  .TYPEO: MOVB    #1,.PR          ;SET ZERO FILL SWITCH
020644  000402                  BR      .+6             ;SKIP
020646  005067  000142          .TYPES: CLR     .PR             ;SUPRESS LEADING ZERO'S
020652  112767  177772  000135          MOVB    #-6,.PR+1       ;SET COUNT
020660                          .PTIT:
020660  010446                          MOV     R4,-(6)         ;PUSH R4 ON STACK
020662  010546                          MOV     R5,-(6)         ;PUSH R5 ON STACK
020664  016605  000010                  MOV     10(6),R5        ;GET THE DATA
020670  012704  021016                  MOV     #.PR+2,R4       ;SET POINTER TO FIRST ASCII CHAR.
020674  105014                          CLRB    (4)             ;CLEAR FIRST BYTE
020676  000411                          BR      .PRF            ;ROTATE FIRST BIT
020700  105014                  .PRL:   CLRB    (4)             ;CLEAR BYTE OF CHARACTER
020702  032767  000100  000104          BIT     #100,.PR        ;BIT TYPING MODE?
020710  001004                          BNE     .PRF            ;YES - SKIP 2 ROTATES
020712  006105                          ROL     R5              ;ROTATE BIT INTO C
020714  106114                          ROLB    (4)             ;PACK IT
020716  006105                          ROL     R5              ;ROTATE BIT INTO C
020720  106114                          ROLB    (4)             ;PACK IT
020722  006105                  .PRF:   ROL     R5              ;ROTATE BIT INTO C
020724  106114                          ROLB    (4)             ;PACK IT
020726  105714                          TSTB    (4)             ;IS IT ZERO?
020730  001402                          BEQ     .+6             ;SKIP INC
020732  105267  000056                  INCB    .PR             ;SET FILL SWITCH
020736  105767  000052                  TSTB    .PR             ;CHECK FILL SWITCH
020742  001402                          BEQ     .+6             ;SKIP BITSET
020744  152724  000060                  BISB    #'0,(4)+        ;MAKE INTO ASCII CHAR
020750  105267  000041                  INCB    .PR+1           ;INC COUNT
020754  001351                          BNE     .PRL            ;REPEAT
020756  022704  021016                  CMP     #.PR+2,R4       ;EMPTY BUFFER?
020762  001002                          BNE     .+6             ;SKIP IF NOT
020764  112724  000060                  MOVB    #'0,(4)+        ;LOAD 1 ZERO
020770  105014                          CLRB    (4)             ;NULL TERMINATOR
020772  104402  021016                  TYPE    .PR+2           ;TYPE IT
020776  012605                          MOV     (6)+,R5         ;POP STACK INTO R5
021000  012604                          MOV     (6)+,R4         ;POP STACK INTO R4
021002  016666  000002  000004          MOV     2(6),4(6)       ;GET RID OF
021010  012616                          MOV     (6)+,(6)        ;DATA WORD
021012  000002                          RTI                     ;RETURN

021014  000012          .PR:    .BLKW   12              ;COUNT, SWITCH, AND OUTPUT BUFFER
```

```
                                        .SBTTL          $POWER - POWER DOWN AND UP ROUTINES
                                        ;THIS IS THE POWER FAIL ROUTINE WHICH WILL SAVE ALL
                                        ;THE GENERAL REGISTERS AND USER DEFINED REGISTERS THEN
                                        ;WAIT FOR POWER TO GO DOWN AND BE RESTORED.
                                        ;IF THERE ISN'T ENOUGH TIME FOR SAVING ALL THE REGISTERS,
                                        ;THE PROGRAM WILL HALT AT '.ILLUP'.

021040  012777  021164  000124  .POWER: MOV     #.ILLUP,@.PUVEC ;SET FOR FAST UP
021046  012777  000340  000120          MOV     #340,@.PUVEC$+2 ;PRIO:7
021054  010046                          MOV     R0,-(6)         ;PUSH R0 ON STACK
021056  010146                          MOV     R1,-(6)         ;PUSH R1 ON STACK
021060  010246                          MOV     R2,-(6)         ;PUSH R2 ON STACK
021062  010346                          MOV     R3,-(6)         ;PUSH R3 ON STACK
021064  010446                          MOV     R4,-(6)         ;PUSH R4 ON STACK
021066  010546                          MOV     R5,-(6)         ;PUSH R5 ON STACK
021070  010667  000074                  MOV     SP,.SAVR6       ;SAVE SP
021074  012777  021104  000070          MOV     #.POWUP,@.PUVEC ;SET UP VECTOR
021102  000000                          HALT                    ;WAIT FOR PF

021104  016706  000060  .POWUP: MOV     .SAVR6,SP       ;GET SP
021110  005001                  CLR     R1              ;WAIT LOOP FOR THE TTY
021112  005201          1$:     INC     R1              ;WAIT FOR THE INC
021114  001376                  BNE     1$              ;OF  WORD
021116  012605                  MOV     (6)+,R5         ;POP STACK INTO R5
021120  012604                  MOV     (6)+,R4         ;POP STACK INTO R4
021122  012603                  MOV     (6)+,R3         ;POP STACK INTO R3
021124  012602                  MOV     (6)+,R2         ;POP STACK INTO R2
021126  012601                  MOV     (6)+,R1         ;POP STACK INTO R1
021130  012600                  MOV     (6)+,R0         ;POP STACK INTO R0
021132  012737  021040  000024          MOV     #.POWER,@#24    ;SET UP THE POWER DOWN VECTOR
021140  012737  000340  000026          MOV     #340,@#26       ;PRIO:7
021146  104402  021152                  TYPE    ,.+2            ;.ASCIZ <15><12>"POWER"
021162  000002                          RTI                     ;RETURN

021164  000000  .ILLUP: HALT                    ;THE POWER UP SEQUENCE WAS STARTED
021166  000776          BR      .-2             ; BEFORE THE POWER DOWN WAS COMPLETE

021170  000000  .SAVR6: 0                       ;PUT THE SP HERE
021172  000024  000026  .PUVEC: 24,26           ;POWER UP VECTOR
```

```
                                        .SBTTL          $RDOCT - OCTAL INPUT ROUTINE

                                        ;THIS ROUTINE CALLS RDLIN, INPUTS A LINE FROM THE TTY AND CONVERTS
                                        ;IT INTO AN OCTAL NUMBER WHICH IS THE FIRST WORD ON THE STACK.

021176  011646                  .RDOCT: MOV     (6),-(6)        ;MOVE THE PC
021200  016666  000004  000002          MOV     4(6),2(6)       ;MOVE THE PS
021206  010146                          MOV     R1,-(6)         ;PUSH R1 ON STACK
021210  010246                          MOV     R2,-(6)         ;PUSH R2 ON STACK
021212  010346                          MOV     R3,-(6)         ;PUSH R3 ON STACK
021214  104412                  4$:     RDLIN                   ;READ A LINE INTO INPUT
021216  005001                          CLR     R1              ;INIT DATA WORD
021220  012703  021420                  MOV     #INPUT,R3       ;INIT POINTER
021224  112302                  1$:     MOVB    (3)+,R2         ;GET A BYTE
021226  001417                          BEQ     2$              ;GET OUT IF ZERO
021230  122702  000060                  CMPB    #'0,R2          ;CHECK FOR 0 OR GREATER
021234  003022                          BGT     3$              ;ERROR - LESS THAN 0
021236  122702  000067                  CMPB    #'7,R2          ;CHECK FOR 7 OR LESS
021242  002417                          BLT     3$              ;ERROR - GREATER THAN 7
021244  006002                          ROR     R2              ;GET
021246  006002                          ROR     R2              ;INTO
021250  006002                          ROR     R2              ;POSITION
021252  006101                          ROL     R1              ;FIRST BIT
021254  006102                          ROL     R2              ;GET
021256  006101                          ROL     R1              ;SECOND BIT
021260  006102                          ROL     R2              ;GET
021262  006101                          ROL     R1              ;THIRD BIT
021264  000757                          BR      1$              ;LOOP
021266  010166  000012          2$:     MOV     R1,12(6)        ;SAVE THE RESULT
021272  012603                          MOV     (6)+,R3         ;POP STACK INTO R3
021274  012602                          MOV     (6)+,R2         ;POP STACK INTO R2
021276  012601                          MOV     (6)+,R1         ;POP STACK INTO R1
021300  000002                          RTI                     ;RETURN

021302                          3$:
021302  104402  021306                  TYPE    ,.+2            ;.ASCIZ "?"<15><12>
021312  000740                          BR      4$              ;TRY AGAIN
```

```
                                .SBTTL          $RDLIN - TTY INPUT ROUTINE

                                ;THIS ROUTINE INPUTS A LINE TERMINATED BY A RETURN INTO ADDRESS
                                ;INPUT AND RETURNS A LINE FEED. THE BUFFER HAS A NULL TERMINATOR
                                ;INSTEAD OF THE RETURN. RUBOUTS ARE HANDLED BY RETYPING
                                ;THE LINE. BUFFER OVERFLOW ERRORS LIKE A RUBOUT.

021314  010546                  .RDLIN: MOV     R5,-(6)         ;SAVE R5
021316  012705  021420          1$:     MOV     #INPUT,R5       ;GET ADDRESS
021322  022705  021440          2$:     CMP     #INPUT+16.,R5   ;BUFFER FULL?
021326  001412                          BEQ     4$              ;YES - TYPE "?"
021330  105737  177560                  TSTB    @#177560        ;WAIT FOR
021334  100375                          BPL     .-4             ;A CHARACTER
021336  113715  177562                  MOVB    @#177562,(5)    ;GET CHARACTER
021342  142715  000200                  BICB    #200,(5)        ;GET RID OF JUNK
021346  122715  000177                  CMPB    #177,(5)        ;IS IT A RUBOUT
021352  001005                          BNE     3$              ;SKIP IF NOT
021354                          4$:
021354  104402  021360                  TYPE    .+2             ;.ASCIZ "?"<15><12>
021364  000754                          BR      1$              ;ZAP THE BUFFER AND LOOP
021366  111527  000000          3$:     MOVB    (5),#0          ;SET UP FOR TYPING
021372  104402  021370                  TYPE    3$+2            ;ECHO IT
021376  122725  000015                  CMPB    #15,(5)+        ;CHECK FOR RETURN
021402  001347                          BNE     2$              ;LOOP IF NOT RETURN
021404  105065  177777                  CLRB    -1(5)           ;ZAP RETURN (THE 15)
021410  104402  000012                  TYPE    ,12             ;TYPE A LINE FEED
021414  012605                          MOV     (6)+,R5         ;RESTORE R5
021416  000002                          RTI                     ;RETURN

021420  000020          INPUT:  .BLKB   16.                     ;TTY INPUT AREA
                        .SBTTL          $TRAP - TRAP HANDLER

                                ;THIS ROUTINE DECODES A TRAP CALL AND JUMPS TO THE APROPRATE
                                ;SUBROUTINE. THE CALL IS A "TRAP+N" WHERE N IS A MULTIPLE OF 2.
                                ;THE "SET" MACRO WILL CREATE THE TABLE NEEDED. IT HAS TO
                                ;FOLLOW THIS MACRO.

021440  011646          .TRAP:  MOV     (6),-(6)        ;GET ADDRESS OF TRAP +2
021442  162716  000002          SUB     #2,(6)          ;MAKE IT ADDRESS OF TRAP
021446  017616  000000          MOV     @(6),(6)        ;GET TRAP INSTRUCTION
021452  062716  115060          ADD     #.TRP+2-TRAP,(6);GET DATA AND MAKE IT AN OFFSET
021456  013607          .TRP:   MOV     @(6)+,PC        ;GO TO PROPER SUBROUTINE

                                .SCOPE                  ;SCOPE  =  TRAP+0       (104400)
021460  020276                  .TYPE                   ;TYPE   =  TRAP+2       (104402)
021462  020140                  .TYPEO                  ;TYPEO  =  TRAP+4       (104404)
021464  020636                  .TYPES                  ;TYPES  =  TRAP+6       (104406)
021466  020646                  .RDOCT                  ;RDOCT  =  TRAP+10      (104410)
021470  021176                  .RDLIN                  ;RDLIN  =  TRAP+12      (104412)
021472  021314                  .CLRDK                  ;CLRDK  =  TRAP+14      (104414)
021474  023526
```

```
                                        ;ROUTINE TO ALLOW THE OPERATOR TO SET BITS
                                        ;IN THE I/O REGISTERS VIA THE SWITCH REGISTER

                                        ;WORD COUNT REGISTER
021476  016777  156066  157404  SRSWC:  MOV     SWR,@RSWC       ;MOV SWR INTO WORD COUNT REG
021504  017737  157400  177570          MOV     @RSWC,@#DISPLAY ;DISPLAY IN LIGHTS
021512  000771                          BR      SRSWC

                                        ;CURRENT ADDRESS REGISTER
021514  016777  156050  157370  SRSBA:  MOV     SWR,@RSBA       ;MOV SWR INTO CURRENT ADDR REG
021522  017737  157364  177570          MOV     @RSBA,@#DISPLAY ;SHOW IN LIGHTS
021530  000771                          BR      SRSBA

                                        ;DISK ADDRESS REGISTER
021532  016777  156032  157354  SRSDA:  MOV     SWR,@RSDA       ;MOV SWR INTO DISK ADDR REG
021540  017737  157350  177570          MOV     @RSDA,@#DISPLAY ;SHOW IN LIGHTS
021546  000771                          BR      SRSDA

                                        ;DRIVE STATUS REGISTER
021550  016777  156014  157340  SRSDS:  MOV     SWR,@RSDS       ;MOV SWR INTO DRIVE STATUS
021556  017737  157334  177570          MOV     @RSDS,@#DISPLAY ;SHOW IN LIGHTS
021564  000771                          BR      SRSDS

                                        ;DRIVE ERROR REGISTER
021566  016777  155776  157324  SRSER:  MOV     SWR,@RSER       ;LOAD ER REG
021574  017737  157320  177570          MOV     @RSER,@#DISPLAY ;DISPLAY IT IN LIGHTS
021602  000771                          BR      SRSER           ;LOOP

                                        ;WATCH LOOK AHEAD REGISTER
021604  017737  157314  177570  SRSLA:  MOV     @RSLA,@#DISPLAY ;SHOW IN LIGHTS
021612  000774                          BR      SRSLA
```

```
 4664                                          ;RSCS2 REGISTER
 4665  021614  016777  155750  157264  $RCS2:  MOV     $WR,@RSCS2      ;LOAD CS2
 4666  021622  017737  157260  177570          MOV     @RSCS2,@#DISPLAY        ;DISPLAY IT
 4667  021630  000771                          BR      $RCS2
 4668
 4669                                          ;RSAS REGISTER
 4670  021632  016777  155732  157262  $RAS:   MOV     $WR,@RSAS       ;LOAD RSAS
 4671  021640  017737  157256  177570          MOV     @RSAS,@#DISPLAY ;DISPLAY IT
 4672  021646  000771                          BR      $RAS
 4673
 4674                                          ;RSMR REGISTER
 4675  021650  016777  155714  157252  $RSMRR: MOV     $WR,@RSMR       ;LOAD RSMR
 4676  021656  017737  157246  177570          MOV     @RSMR,@#DISPLAY ;DISPLAY IT
 4677  021664  000771                          BR      $RSMRR
 4678
 4679                                          ;DISK CONTROL STATUS REGISTER
 4680  021666  012737  000340  177776  $RSCS1: MOV     #340,@#PS       ;LOCK UP INTERRUPTS
 4681  021674  012777  177777  157206          MOV     #177777,@RSWC   ;SET WORD COUNT -1 WORD
 4682  021702  013777  001102  157202          MOV     @#OBUFSV,@RSBA  ;SET UP CURRENT ADDRESS
 4683  021710  016777  155654  157166          MOV     $WR,@RSCS1      ;MOV SWR INTO CONTROL REG
 4684  021716  032777  000001  157160          BIT     #BIT0,@RSCS1    ;IS FUNCTION BITS SET
 4685  021724  001760                          BEQ     $RSCS1          ;FUNCTION BITS NOT SET
 4686  021726  105777  157152          DKBUSY: TSTB    @RSCS1          ;TEST FOR DISK READY
 4687  021732  100375                          BPL     DKBUSY          ;DISK STILL NOT READY
 4688  021734  000754                          BR      $RSCS1          ;DISK NOT BUSY SECT NEW CR
```

```
                                        ;THIS ROUTINE GIVES THE OPERATOR THE ABILITY TO
                                        ;SELECT DA, WC, UNIT # AND DESIRED PATTERN. PATTERN = NUMBER TYPED
                                        ;WITH SW12 SET THE PROGRAM WILL LOOP ON A READ WITH LOC OUTBUF+2 AS
                                        ;THE BA ADDR. WITH BIT12 0 IN THE SWR THE PROGRAM
                                        ;WILL WRITE WITH OUTBUF AS THE BA ADDR. BAI IS ALWAYS SET
                                        ;SWITCHES 0 TO 11 WILL DETERMINE THE DA
                                        ;EXAMPLE:       TYPE UNIT # 5
                                        ;               TYPE POSITIVE (OCTAL) WC 64
                                        ;               TYPE PATTERN DESIRED 1252525

021736  012706  000500          TKSEL:  MOV     #500,SP         ;SET STACK
021742  000240                          NOP
021744  104402  021750                  TYPE    ,.+2            ;.ASCIZ <15><12>"TYPE UNIT # "
021770  104410                          RDOCT
021772  012667  157160                  MOV     (6)+,UNNUM
021776  104402  022002                  TYPE    ,.+2            ;.ASCIZ <15><12>"TYPE POSITIVE (OCTAL) WC "
022036  104410                          RDOCT
022040  012667  157134                  MOV     (6)+,WORK
022044  005167  157130                  COM     WORK
022050  104402  022054                  TYPE    ,.+2            ;.ASCIZ <15><12>"TYPE PATTERN DESIRED "
022104  104410                          RDOCT
022106  012667  002436                  MOV     (6)+,OUTBUF
022112  016767  155452  157064  TK1:    MOV     SWR,WORK2       ;SAVE SWR
022120  104414                  TK2:    CLRDK                   ;CLEAR ALL RS REG
022122  052777  000010  156756          BIS     #BIT3,@RSCS2    ;SET BAI
022130  016767  155434  157044          MOV     SWR,WORK1       ;GET SWR FOR DSK ADDR
022136  042767  170000  157036          BIC     #170000,WORK1   ;CLEAR UNIT #
022144  016777  157032  156742  TKS:    MOV     WORK1,@RSDA     ;LOAD THE DA
022152  016777  157022  156730          MOV     WORK,@RSWC      ;LOAD WORD COUNT
022160  032767  010000  155402          BIT     #BIT12,SWR      ;READ?
022166  001412                          BEQ     WTE             ;NO
022170  012777  024552  156714          MOV     #OUTBUF+2,@RSBA ;LOAD CURRENT ADDRESS
022176  012777  000071  156700          MOV     #71,@RSCS1      ;GO AND READ
022204  105777  156674          WT:     TSTB    @RSCS1          ;TEST FOR READY
022210  100375                          BPL     .-4
022212  000407                          BR      SWRCHG
022214  012777  024550  156670  WTE:    MOV     #OUTBUF,@RSBA
022222  012777  000061  156654          MOV     #61,@RSCS1
022230  000765                          BR      WT
022232  017737  156646  177570  SWRCHG: MOV     @RSCS1,@#DISPLAY        ;DISPLAY CS1
022240  005777  156640                  TST     @RSCS1          ;ANY ERRORS?
022244  100001                          BPL     1$              ;NO
022246  104014                          HLT     !DA!WC
022250  026767  155314  156726  1$:     CMP     SWR,WORK2       ;DID SWR CHANGE?
022256  001315                          BNE     TK1             ;YES
022260  000717                          BR      TK2             ;NO
```

```
                                ;*****************************************************************************
                                ;TEST 112                   WRITE LOCK TEST
                                ;*****************************************************************************
022262  104400                  TST112: SCOPE

022264                          WRTLCK:
022264  104402  022270                  TYPE    ,.+2            ;.ASCIZ <15><12>"LOAD SW WITH UNIT # AND CONT"
022330  000000                          HALT
022332  013767  177570  156616        MOV     @#SWR,UNNUM     ;GET UNIT #
022340  104414                        CLRDK                   ;CLEAR ALL REG
022342  005067  002202                CLR     OUTBUF          ;PUT A 0 INTO DATA BUFFER
022346  012777  024550  156536        MOV     #OUTBUF,@RSBA   ;SETUP REG TO
022354  012777  007700  156532        MOV     #7700,@RSDA     ;TO A WRITE
022362  012777  177777  156520        MOV     #-1,@RSWC
022370  012777  000061  156506        MOV     #61,@RSCS1
022376  105777  156502          6$:   TSTB    @RSCS1          ;WAIT FOR DONE
022402  100375                        BPL     6$
022404  104402  022410                TYPE    ,.+2            ;.ASCIZ <15><12>"SET WRITE LOCK ENABLE AND CONTINUE"
022456  000000                        HALT
022460  012777  000000  156426  1$:   MOV     #0,@RSDA
022466  022777  014600  156422        CMP     #14600,@RSDS
022474  001401                        BEQ     .+4
022476  104044                        HLT     !DS!DA          ;DS SHOULD=14600
022500  012777  000100  156406        MOV     #100,@RSDA
022506  022777  010600  156402        CMP     #10600,@RSDS
022514  001401                        BEQ     .+4
022516  104044                        HLT     !DS!DA          ;DS SHOULD=10600
022520  104402  022524                TYPE    ,.+2            ;.ASCIZ <15><12>"SET WRT LOC SW 0 AND CONT"
022560  000000                        HALT
022562  022777  014600  156326  2$:   CMP     #14600,@RSDS
022570  001401                        BEQ     .+4
022572  104044                        HLT     !DS!DA          ;DS SHOULD=14600
022574  012777  000300  156312        MOV     #300,@RSDA
022602  022777  010600  156306        CMP     #10600,@RSDS
022610  001401                        BEQ     .+4
022612  104044                        HLT     !DS!DA          ;DS SHOULD=10600
022614  104402  022620                TYPE    ,.+2            ;.ASCIZ <15><12>"SET WRT LOC SW 1 AND CONT"
022654  000000                        HALT
022656  022777  014600  156232  3$:   CMP     #14600,@RSDS
022664  001401                        BEQ     .+4
022666  104044                        HLT     !DS!DA          ;DS SHOULD=14600
022670  012777  000700  156216        MOV     #700,@RSDA
022676  022777  010600  156212        CMP     #10600,@RSDS
022704  001401                        BEQ     .+4
022706  104044                        HLT     !DS!DA          ;DS SHOULD=10600
022710  104402  022714                TYPE    ,.+2            ;.ASCIZ <15><12>"SET WRT LCK SW 2 AND CONT"
022750  000000                        HALT
```

```
 4782  022752  022777  014600  156136  4$:     CMP     #14600,@RSDS
 4783  022760  001401                          BEQ     .+4
 4784  022762  104044                          HLT     !DS!DA          ;DS SHOULD=14600
 4785  022764  012777  001700  156122          MOV     #1700,@RSDA
 4786  022772  022777  010600  156116          CMP     #10600,@RSDS
 4787  023000  001401                          BEQ     .+4
 4788  023002  104044                          HLT     !DS!DA          ;DS SHOULD=10600
 4789  023004  104402  023010                  TYPE    ,.+2            ;.ASCIZ <15><12>"SET WRT LCK SW 3 AND CONT"
 4790  023044  000000          HALT:           HALT
 4791  023046  022777  014600  156042          CMP     #14600,@RSDS
 4792  023054  001401                          BEQ     .+4
 4793  023056  104044                          HLT     !DS!DA          ;DS SHOULD=14600
 4794  023060  012777  003700  156026          MOV     #3700,@RSDA
 4795  023066  022777  010600  156022          CMP     #10600,@RSDS
 4796  023074  001401                          BEQ     .+4
 4797  023076  104044                          HLT     !DS!DA          ;DS SHOULD=10600
 4798  023100  104402  023104                  TYPE    ,.+2            ;.ASCIZ <15><12>"SET WRT LCK SW 4 AND CONT"
 4799  023140  000000                          HALT
 4800  023142  022777  014600  155746          CMP     #14600,@RSDS
 4801  023150  001401                          BEQ     .+4
 4802  023152  104044                          HLT     !DS!DA          ;DS SHOULD=14600
 4803  023154  012777  006000  155732          MOV     #6000,@RSDA
 4804  023162  022777  010600  155726          CMP     #10600,@RSDS
 4805  023170  001401                          BEQ     .+4
 4806  023172  104044                          HLT     !DS!DA          ;DS SHOULD=10600
 4807  023174  104402  023200                  TYPE    ,.+2            ;.ASCIZ <15><12>"SET WRT LCK SW 5 AND CONT"
 4808  023234  000000                          HALT
 4809  023236  022777  014600  155652          CMP     #14600,@RSDS
 4810  023244  001401                          BEQ     .+4
 4811  023246  104044                          HLT     !DS!DA          ;DS SHOULD=14600
 4812  023250  012777  007700  155636          MOV     #7700,@RSDA
 4813  023256  022777  014600  155632          CMP     #14600,@RSDS
 4814  023264  001401                          BEQ     .+4
 4815  023266  104044                          HLT     !DS!DA          ;DS SHOULD=14600
 4816  023270  012767  177777  001252          MOV     #-1,OUTBUF      ;PUT A 1 INTO DATA BUFFER
 4817  023276  012777  024550  155606          MOV     #OUTBUF,@RSBA   ;SETUP REG TO
 4818  023304  012777  007700  155602          MOV     #7700,@RSDA     ;TO A WRITE
 4819  023312  012777  177777  155570          MOV     #-1,@RSWC
 4820  023320  012777  000061  155556          MOV     #61,@RSCS1      ;TRY TO WRITE
 4821  023326  105777  155552          7$:     TSTB    @RSCS1          ;WAIT FOR DONE
 4822  023332  100375                          BPL     7$
 4823  023334  105777  155544          5$:     TSTB    @RSCS1          ;WAIT FOR READY
 4824  023340  100375                          BPL     5$
 4825  023342  022777  154600  155546          CMP     #154600,@RSDS
 4826  023350  001401                          BEQ     .+4
 4827  023352  104044                          HLT     !DS!DA          ;DS SHOULD=154600
```

```
 4828  023354  022777  004000  155536          CMP     #4000,@RSER
 4829  023362  001401                          BEQ     .+4
 4830  023364  104044                          HLT     !DS!DA          ;ER SHOULD=4000
 4831  023366  104414                          CLRDK                   ;CLEAR ALL REG
 4832  023370  022777  014600  155520          CMP     #14600,@RSDS    ;DID WLE CLEAR?
 4833  023376  001401                          BEQ     .+4             ;NO
 4834  023400  104040                          HLT     !DS             ;A CLEAR SHOULD NOT CLEAR WLE
 4835  023402  005067  001142                  CLR     OUTBUF          ;PUT A 0 INTO DATA BUFFER
 4836  023406  012777  024550  155476          MOV     #OUTBUF,@RSBA   ;SETUP REG TO
 4837  023414  012777  007700  155472          MOV     #7700,@RSDA     ;TO A WRITE
 4838  023422  012777  177777  155460          MOV     #-1,@RSWC
 4839  023430  012777  000071  155446          MOV     #71,@RSCS1      ;DO A READ TO SEE IF DISK DID
 4840  023436  105777  155442          8$:     TSTB    @RSCS1          ;ACTUALLY GET WRITTEN ON TO
 4841  023442  100375                          BPL     8$              ;WAIT FOR DONE
 4842  023444  005767  001100                  TST     OUTBUF          ;IS DATA STILL 0 ON THE DSK?
 4843  023450  001401                          BEQ     .+4             ;YES
 4844  023452  104040                          HLT     !DS             ;NO DSK DID GET WRITTEN ONTO WITH WLE SET
 4845  023454  104402  023460                  TYPE    ,.+2            ;.ASCIZ <15><12>"RESET ALL W/L SWITCHES"
 4846  023512  000000                          HALT
 4847  023514  022777  010600  155374          CMP     #10600,@RSDS    ;DID WLE CLEAR
 4848  023522  001401                          BEQ     .+4             ;YES
 4849  023524  104040                          HLT     !DS             ;NO
 4850                          ;CLEAR ALL DISK REGISTERS
 4851  023526  012777  000040  155352  .CLRDK: MOV     #40,@RSCS2      ;CLEAR ALL DSK REG
 4852  023534  016777  155416  155344          MOV     UNNUM,@RSCS2    ;GET UNIT NUMBER
 4853  023542  000002                          RTI
```

```
                                        ;ERROR TYPTXTOUT ROUTINE
                                RSREG:  TST     .HLTCT          ;SHOULD WE TYPTXT GOOD AND BAD
                                        BNE     8$              ;NO
                                        TYPE    .+2             ;.ASCIZ " BAD="
                                        MOV     @.+2,-(6)       ;PUT BAD ON STACK
                                        TYPEO                   ;TYPE STACK IN OCTAL
                                        TYPE    .+2             ;.ASCIZ " GOOD="
                                        MOV     @.+2,-(6)       ;PUT GOOD ON STACK
                                        TYPEO                   ;TYPE STACK IN OCTAL
                                        JMP     PTDONE          ;GET OUT
                                8$:
                                        TYPE    .+2             ;.ASCIZ " CS1="
                                        MOV     @#RSCS1,-(6)    ;PUT @#RSCS1 ON STACK
                                        TYPEO                   ;TYPE STACK IN OCTAL
                                1$:
                                        TYPE    .+2             ;.ASCIZ " ER="
                                        MOV     @#RSER,-(6)     ;PUT @#RSER ON STACK
                                        TYPEO                   ;TYPE STACK IN OCTAL
                                2$:
                                        TYPE    .+2             ;.ASCIZ " CS2="
                                        MOV     @#RSCS2,-(6)    ;PUT @#RSCS2 ON STACK
                                        TYPEO                   ;TYPE STACK IN OCTAL
                                        BIT     #200,.HLTCT     ;TYPTXT SECOND SET ?
                                        BNE     SEEC            ;YES
                                        BIT     #AS,.HLTCT      ;TYPTXT ER ?
                                        BEQ     3$              ;NO
                                        TYPE    .+2             ;.ASCIZ " AS="
                                        MOV     @#RSAS,-(6)     ;PUT @#RSAS ON STACK
                                        TYPEO                   ;TYPE STACK IN OCTAL
                                3$:     BIT     #BA,.HLTCT      ;TYPTXT BUS ASSRESS
                                        BEQ     4$              ;NO
                                        TYPE    .+2             ;.ASCIZ " BA="
                                        MOV     @#RSBA,-(6)     ;PUT @#RSBA ON STACK
                                        TYPEO                   ;TYPE STACK IN OCTAL
                                4$:     BIT     #DA,.HLTCT      ;TYPTXT DA ?
                                        BEQ     5$              ;NO
                                        TYPE    .+2             ;.ASCIZ " DA="
                                        MOV     @#RSDA,-(6)     ;PUT @#RSDA ON STACK
                                        TYPEO                   ;TYPE STACK IN OCTAL
                                5$:     BIT     #WC,.HLTCT      ;TYPTXT WC?
                                        BEQ     6$              ;NO
                                        TYPE    .+2             ;.ASCIZ " WC="
                                        MOV     @#RSWC,-(6)     ;PUT @#RSWC ON STACK
                                        TYPEO                   ;TYPE STACK IN OCTAL
```

```
4899  024044  032767  000040  174552  6$:     BIT     #DS,.HLTCT      ;DRIVE STATUS
4900  024052  001475                          BEQ     PTDONE          ;NO
4901  024054  104402  024060                  TYPE    .+2             ;.ASCIZ " DS="
4902  024066  017746  155024                  MOV     @RSDS,-(6)      ;PUT @RSDS ON STACK
4903  024072  104404                          TYPEO                   ;TYPE STACK IN OCTAL
4904  024074  000167  000146                  JMP     PTDONE          ;GET OUT
4905  024100  042767  000200  174516  SEEC:   BIC     #200,.HLTCT     ;CLEAR COMMON BIT
4906  024106  032767  000240  174510          BIT     #DT,.HLTCT      ;TYPTXT DRIVE TYPE?
4907  024114  001410                          BEQ     9$              ;NO
4908  024116  104402  024122                  TYPE    .+2             ;.ASCIZ " DT="
4909  024130  017746  154776                  MOV     @RSDT,-(6)      ;PUT @RSDT ON STACK
4910  024134  104404                          TYPEO                   ;TYPE STACK IN OCTAL
4911  024136  032767  000210  174460  9$:     BIT     #DB,.HLTCT      ;TYPTXT DATA BUFFER
4912  024144  001410                          BEQ     10$             ;NO
4913  024146  104402  024152                  TYPE    .+2             ;.ASCIZ " DB="
4914  024160  017746  154742                  MOV     @RSDB,-(6)      ;PUT @RSDB ON STACK
4915  024164  104404                          TYPEO                   ;TYPE STACK IN OCTAL
4916  024166  032767  000220  174430  10$:    BIT     #MR,.HLTCT      ;TYPTXT MN?
4917  024174  001410                          BEQ     11$             ;NO
4918  024176  104402  024202                  TYPE    .+2             ;.ASCIZ " MR="
4919  024210  017746  154714                  MOV     @RSMR,-(6)      ;PUT @RSMR ON STACK
4920  024214  104404                          TYPEO                   ;TYPE STACK IN OCTAL
4921  024216  032767  000204  174400  11$:    BIT     #LA,.HLTCT      ;TYPTXT LA?
4922  024224  001410                          BEQ     PTDONE          ;NO
4923  024226  104402  024232                  TYPE    .+2             ;.ASCIZ " LA="
4924  024240  017746  154660                  MOV     @RSLA,-(6)      ;PUT @RSLA ON STACK
4925  024244  104404                          TYPEO                   ;TYPE STACK IN OCTAL
4926  024246  052767  100000  154710  PTDONE: BIS     #BIT15,ONCEE    ;SET FORND ERROR FLAG
4927  024254  000207                          RTS     PC
4928
4929  024256  005067  154716          WAITRY: CLR     WORK            ;CLEAR COUNTER
4930  024262  105777  154616          1$:     TSTB    @RSCS1          ;TEST READY
4931  024266  100406                          BMI     2$              ;OK CONT
4932  024270  005267  154704                  INC     WORK            ;UPDATE COUNTER
4933  024274  005767  154700                  TST     WORK            ;DONE YET?
4934  024300  001403                          BEQ     3$              ;READY DID NOT COME UP
4935  024302  000767                          BR      1$              ;CONTINUE WAITING
4936  024304  062716  000002          2$:     ADD     #2,(SP)         ;UPDATE RETURN PC
4937  024310  000207                  3$:     RTS     PC              ;RETURN
```

```
                                        ;RANDOM DATA GENERATOR SUBROUTINE
                                        ;
024312  016767  000136  000140  RANDOM: MOV     LONUM,LOSAV
024320  016767  000132  000134          MOV     HINUM,HISAV
024326  016700  000122          RAND1:  MOV     LONUM,R0        ;SET UP R0 WITH 5 DIGITS LOW
024332  016704  000120                  MOV     HINUM,R4        ;SET UP R1 WITH 5 DIGITS HIGH
024336  012703  000007                  MOV     #7,R3           ;SET UP SHIFT COUNT
024342  005002                          CLR     R2              ;CLEAR R2
024344  006300                  SHIFT:  ASL     R0              ;SHIFT R0 LEFT AND
024346  006104                          ROL     R4              ;ROTATE CARRY INTO LSB OF R1 INTO
024350  006102                          ROL     R2              ;ROTATE CARRY OUT OF R1 INTO R2
024352  005303                          DEC     R3              ;DECREMENT R3
024354  001373                          BNE     SHIFT           ;CONTINUE SHIFT LOOP
024356  066700  000072                  ADD     LONUM,R0        ;ADDN IN NUMBER TO MAKE X 129
024362  005504                          ADC     R4              ;PROPOGATE CARRY
024364  066704  000066                  ADD     HINUM,R4        ;ADDN IN NUMBER TO MAKE X 129
024370  005502                          ADC     R2              ;PROPOGATE CARRY
024372  062700  001057                  ADD     #1057,R0        ;ADDN LOW CONSTANT
024376  005504                          ADC     R4              ;PROPOGATE CARRIES
024400  005502                          ADC     R2              ;PROPOGATE AGAIN
024402  062704  047401                  ADD     #47401,R4       ;ADDN HIGH CONSTANT
024406  005502                          ADC     R2              ;PROPOGATE CARRY
024410  062702  000006                  ADD     #6,R2           ;ADDN HIGHEST CONSTANT
024414  062700  000002                  ADD     #2,R0           ;REPRIME R0 WITH HIGH DIGIT
024420  005504                          ADC     R4              ;PROPOGATE CARRY
024422  010067  000026                  MOV     R0,LONUM        ;PUT R0 BACK IN LONUM
024426  010021                          MOV     R0,(R1)+        ;LOAD WC
024430  005367  154544                  DEC     WORK
024434  001406                          BEQ     EXGEN
024436  010467  000014                  MOV     R4,HINUM        ;PUT R1 BACK IN HINUM
024442  010421                          MOV     R4,(1)+         ;HOLD HINUM FOR PROGRAM
024444  005367  154530                  DEC     WORK
024450  001326                          BNE     RAND1
024452  000205                  EXGEN:  RTS     R5              ;RETURN TO PROGRAM
024454  000000                  LONUM:  0
024456  000000                  HINUM:  0
024460  000000                  LOSAV:  0
024462  000000                  HISAV:  0
024464                          RANEND:
```

```
                                ;TABLES FOR ILLEGAL FUNCTION TESTS

024464  000003          ILLTAB: 3
024466  000005                  5
024470  000007                  7
024472  000013                  13
024474  000015                  15
024476  000017                  17
024500  000023                  23
024502  000025                  25
024504  000027                  27
024506  000033                  33
024510  000035                  35
024512  000037                  37
024514  000041                  41
024516  000043                  43
024520  000045                  45
024522  000047                  47
024524  000000                  0

024526  000053          ILFTB2: 53
024530  000055                  55
024532  000057                  57
024534  000063                  63
024536  000065                  65
024540  000073                  73
024542  000075                  75
024544  000077                  77
024546  000000                  0
024550  000300          OUTBUF: .BLKW   300
025350  000300          INBUF:  .BLKW   300
        000001                  .END
```

```
ADDCF   010772          3168    3177#
AS    = 000100          1799#   2003    2444    2454    2476    2507    2520    2542    2585    2879    2937    3108    3344
                        3610    3650    3735    3742    3779    3800    3806    4261    4281    4303    4307    4310    4879
ATA   = 100000          1820#
AOBI    001172          1834#
BA    = 000020          1797#   1973    2031    2037    2119    2123    2127    2149    2388    2392    2441    2473    2504
                        2539    2605    2885    2940    3001    3017    3035    3776    3839    4884
BAD   =%000000          1729#   2096*   2135*   2136    2173*   2174    2210*   2211    2306*   2320*   2335*   2365*   2366
                        2401*   2402    2408*   2409    2413*   2414    2562*   2563*   2566    2626*   2627    2630*   2633
                        2667*   2668    2671*   2674    2677*   2679    2702*   2703    2715*   2718    2761*   2762    2765*
                        2768    2788*   2789    2814*   2815    2822*   2825    2874*   2906*   2910    2911*   2912    2916*
                        2917    2926*   2929    2934*   2935    2962*   2967    2969    2970*   2971    2975*   2976    2993*
                        2996    3007*   3010    3022*   3025    3060*   3062    3067*   3068    3086*   3087    3092*   3093
                        3149*   3150*   3151    3186*   3187*   3189*   3194    3230*   3233    3284*   3287    3290*   3300*
                        3303    3309*   3310    3347*   3348    3359*   3360    3538*   3541    3545*   3546    3555*   3556
                        3640*   3642    3730*   3733    3759*   3762    3791*   3794    3801*   3804    3814*   3817    3861*
                        3864    3896*   3899*   3900    3995*   3996    4059*   4097*   4140*   4162*   4255*   4256*   4258
                        4267*   4268    4271*   4274    4859
BAI   = 000010          1823#   2783    2796    2805    3826    3849    3883    3960
BAITST  006544          2780#
BEGIN   001206          1674    1692    1854#   4339
BELL  = 000007          1703#   4334    4436
BITBA   004322          2384#
BITCS2  004206          2360#
BITST   004134          2347#
BITWC   004252          2372#
BIT0  = 000001          1712#   1920    1938    2332    2606    2656    2709    2911    2970    4684
BIT1  = 000002          1713#
BIT10 = 002000          1722#
BIT11 = 004000          1673    1691    1723#   1865
BIT12 = 010000          1724#   3660    3663    4718
BIT13 = 020000          1725#   1884    1904    1924
BIT14 = 040000          1665    1726#   3267    3292    3316    3319    3323
BIT15 = 100000          1727#   1887    1906    1913    4256    4926
BIT2  = 000004          1714#
BIT3  = 000010          1715#   4225    4231    4286    4713
BIT4  = 000020          1716#   3789    3830    3855    3890
BIT5  = 000040          1717#
BIT6  = 000100          1718#   3899
BIT7  = 000200          1719#   3468
BIT8  = 000400          1720#
BIT9  = 001000          1721#   3167    3171
BLOCK   007004          2834#
BPORTT  001174          1837#   2721*
CHCKDV  001622          1916    1920#
CLRDK = 104414          2061    2347    2393    2395    2422    2450    2485    2516    2556    2595    2646    2690    2730
                        2780    2795    2804    2834    2904    2947    2960    3039    3053    3074    3116    3121    3165
                        3221    3260    3306    3320    3329    3371    3395    3408    3426    3433    3451    3457    3482
                        3493    3559    3579    3597    3614    3626    3669    3688    3700    3719    3728    3788    3810
                        3825    3848    3873    3882    3909    3921    3940    3953    4005    4062    4111    4136    4158
                        4186    4209    4632#   4712    4744    4831
CSTST   010722          3163#
CS1   = 000001          1793#   1968    2026    2066    2070    2074    2082    2352    2356    2429    2461    2492    2527
                        2582    2609    2611    2617    2653    2655    2661    2697    2699    2746    2749    2786    2820
                        2863    2885    2922    2925    2940    2943    2946    2981    2984    2987    3001    3004    3017
                        3020    3035    3038    3299    3315    3338    3341    3353    3382    3416    3441    3472    3523
```

```
                    3585    3613    3678    3687    3709    3718    3753    3930    3939    4120    4126    4145    4175
CS2     = 000200    1800#   1960    2022    2092    2102    2106    2111    2406    2827    3362    3647    3654
DA      = 000004    1795#   1979    2194    2198    2202    2223    2438    2470    2501    2536    2571    2614    2617
                    2623    2658    2708    2711    2749    2752    2758    2946    2987    3388    3537    3588    3773
                    3872    3908    4034    4281    4284    4731    4757    4761    4766    4770    4775    4779    4784
                    4788    4793    4797    4802    4806    4811    4815    4827    4830    4699
DAO     = 001000    1821#
DAOTST  013736      3669#
DAOTT   014074      3700#
DB      = 000210    1802#   4911
DCK     = 100000    1822#
DCKTST  012644      3493#
DISPLA= 177570      1702#   4415*   4419*   4638*   4643*   4648*   4653*   4658*   4662*   4666*   4671*   4676*   4728*
DKADR   013276      3579#
DKBUSY  021726      4686#   4687
DLT     = 100000    1815#
DONCS   011110      3181    3201#
DONE    020036      1932    4328#
DONEE   001770      1921    1932#
DONINT  017062      4172    4176#
DRCLR   005336      2556#
DRY     = 000200    1816#   2846    2856
DS      = 000040    1798#   1964    2464    2510    2530    2548    2617    2623    2638    2658    2661    2708    2749
                    2758    2794    2811    2848    2852    2860    2863    2882    2890    2922    2925    2950    2953
                    2981    2984    3042    3045    3077    3102    3105    3120    3127    3199    3238    3299    3356
                    3365    3391    3394    3401    3422    3425    3447    3450    3478    3481    3485    3534    3537
                    3610    3639    3658    3684    3691    3715    3722    3738    3767    3800    3809    3836    3869
                    3872    3876    3905    3908    3912    3915    3936    3943    4034    4171    4193    4197    4261
                    4284    4307    4310    4757    4761    4766    4770    4775    4779    4784    4788    4793    4797
                    4802    4806    4811    4815    4827    4830    4834    4844    4849    4899
DT      = 000240    1804#   4906
DVNUM   001350      1878    1881#
DVTAB   020120      4215    4242    4343#
ER      = 000002    1794#   1985    2043    2231    2235    2242    2248    2255    2432    2479    2495    2545    2574
                    2617    2658    2661    2708    2749    2798    2922    2925    2981    2984    3073    3098    3120
                    3124    3199    3299    3385    3391    3398    3419    3422    3444    3447    3475    3478    3531
                    3562    3607    3617    3681    3694    3712    3746    3758    3782    3813    3860    3895    3933
                    3998    4034    4038    4050    4054    4073    4076    4088    4092    4171    4193    4197    4284
ERINT   017064      4183#
ERR     = 040000    1819#
ERRINT  017154      4185    4195#
ERRORS  001002      1734#   4437*   4457
ERRR    010654      3148    3154#
ERTST   014220      3728#
EXGEN   024452      4966    4971#
EXTRP   016514      4013    4089    4095    4100#
EXTTRP  016522      4008    4102#
EXTTST  016000      4005#
EXTT1   016322      4062#
EXT1    016524      4101    4103#
FILCHR  001014      1738#   4364    4365
FINTST  017170      4194    4199#
FLOTBA  003024      2132#
FLOTDA  003324      2207#
FLOTWC  003164      2170#
FUNDO   015002      3848#
```

DERSAA.P11 CROSS REFERENCE TABLE -- USER SYMBOLS

GOOD =%000001 1728# 2097* 2132* 2134 2136 2139* 2170* 2172 2174 2177* 2207* 2209 2211
2214* 2307* 2321* 2336* 2363* 2364* 2366 2399* 2400* 2402 2407* 2409 2412*
2414 2564* 2565* 2566 2624* 2625* 2627 2631* 2632* 2633 2665* 2666* 2668
2672* 2673* 2674 2678* 2679 2700* 2701* 2703 2716* 2717* 2718 2759* 2760*
2762 2766* 2767* 2768 2787* 2789 2812* 2813* 2815 2823* 2824* 2825 2853*
2854* 2865* 2867* 2872 2876 2912* 2915* 2917 2927* 2928* 2929 2932* 2933*
2935 2971* 2974* 2976 2994* 2995* 2996 3008* 3009* 3010 3023* 3024* 3025
3061* 3062 3065* 3066* 3068 3084* 3087 3090* 3091* 3093 3145* 3146* 3151
3166* 3169* 3192* 3193* 3194 3231* 3232* 3233 3285* 3286* 3287 3291* 3294*
3301* 3302* 3303 3307* 3308* 3310 3345* 3346* 3348 3358* 3360 3539* 3540*
3541 3544* 3546 3549* 3552* 3554* 3556 3628* 3631* 3635 3641* 3642 3731*
3732* 3733 3760* 3761* 3762 3792* 3793* 3794 3802* 3803* 3804 3815* 3816*
3817 3862* 3863* 3864 3897* 3898* 3900 3993* 3994* 3996 4058* 4096* 4149*
4163* 4257* 4258 4265* 4266* 4268 4272* 4273* 4274 4862
HALT 023044 4790#
HINUM 024456 4941 4943 4953 4967* 4973#
HISAV 024462 4941* 4975#
HLT = 104000 1698# 1960 1964 1968 1973 1979 1985 1991 1997 2003 2022 2026 2031
2037 2043 2048 2053 2066 2070 2074 2082 2092 2098 2102 2106 2111
2119 2123 2127 2138 2149 2157 2161 2165 2176 2186 2194 2198 2202
2213 2223 2231 2235 2242 2248 2255 2265 2272 2280 2288 2308 2322
2337 2352 2356 2368 2376 2380 2388 2392 2406 2411 2416 2429 2432
2435 2438 2441 2444 2454 2461 2464 2467 2470 2473 2476 2479 2492
2495 2498 2501 2504 2507 2510 2520 2527 2530 2533 2536 2539 2542
2545 2548 2568 2571 2574 2579 2582 2585 2588 2602 2605 2609 2611
2614 2617 2620 2623 2629 2635 2638 2653 2655 2658 2661 2664 2670
2676 2681 2697 2699 2705 2708 2711 2714 2720 2746 2749 2752 2755
2758 2764 2770 2786 2791 2794 2798 2811 2817 2820 2827 2843 2848
2852 2860 2863 2875 2879 2882 2885 2888 2890 2919 2922 2925 2931
2937 2940 2943 2946 2950 2953 2978 2981 2984 2987 2998 3001 3004
3012 3017 3020 3027 3035 3038 3042 3045 3064 3070 3073 3077 3089
3095 3098 3102 3105 3108 3120 3124 3127 3139 3156 3173 3196 3199
3235 3238 3289 3295 3299 3305 3312 3315 3338 3341 3344 3350 3353
3356 3362 3365 3382 3385 3388 3391 3394 3398 3401 3416 3419 3422
3425 3441 3444 3447 3450 3472 3475 3478 3481 3485 3523 3531 3534
3537 3543 3548 3558 3562 3585 3588 3607 3610 3613 3617 3639 3644
3647 3650 3654 3658 3678 3681 3684 3687 3691 3694 3709 3712 3715
3718 3722 3735 3738 3742 3746 3753 3758 3764 3767 3770 3773 3776
3779 3782 3796 3800 3806 3809 3813 3819 3836 3839 3842 3860 3866
3869 3872 3876 3895 3902 3905 3908 3912 3915 3930 3933 3936 3939
3943 3998 4034 4038 4050 4054 4060 4073 4076 4088 4092 4098 4120
4126 4145 4150 4170 4171 4175 4193 4197 4260 4261 4264 4270 4276
4281 4284 4303 4307 4310 4731 4757 4761 4766 4770 4775 4779 4784
4788 4793 4797 4802 4806 4811 4815 4827 4830 4834 4844 4849
HLTADR 001012 1737# 4441* 4442* 4443 4444 4458
IAERR 013352 3597#
ICNT 001000 1733# 1861* 4324* 4393 4408 4410 4412* 4413* 4415 4418* 4419 4425 4453*
IE = 000100 1824#
ILFDN 007560 2907 2955#
ILFDNE 010220 2964 3047#
ILFTB2 024526 2961 4998#
ILF67 010222 3053#
ILLFUN 007560 2960#
ILLTAB 024464 2905 4980#
ILL51 007304 2902#
INBUF 025350 5008#

```
INCW    013036          3521#
INPUT   021420          4558    4591    4592    4612#
INTDON  016744          4146    4151#
INTR4   016746          4157#
INTR5   016642          4134#
INT112  016736          4137    4149#
INT114  017050          4159    4173#
IR    = 000100          1810#
KIPAR0= 172340          1772#   4015*
KIPAR1= 172342          1773#   4016*
KIPAR2= 172344          1774#   4017*   4063*
KIPAR7= 172356          1775#   4014*
KIPDR0= 172300          1776#   4018*
KIPDR1= 172302          1777#   4019*
KIPDR2= 172304          1778#   4020*
KIPDR7= 172316          1779#   4021*
LA    = 000204          1801#   3139    3156    4921
LAD     001010          1736#   1862*   2298*   2312*   2326*   4414*   4420    4422    4425
LASTSC  015350          3921#
LATDON  010720          3153    3157#
LATST   010576          3137#
LBT   = 002000          1818#
LONUM   024454          4940    4942    4951    4963*   4972#
LOP1    003744          2298    2299#
LOP2    004014          2312    2313#
LOP3    004064          2326    2327#
LOSAV   024460          4940*   4974#
MEMOUT  016530          4104#
MODDON  020006          4321#
MODNUM  017172          4209#
MPRO  = 172100          1836#   3210
MR    = 000220          1803#   1991    2048    2265    2272    2280    2288    2579    2843    3173    3199    4916
MULTII  001320          1874#
N     = 000113          1697#   1869    1873#   1941    1945#   2004    2008#   2054    2058#   2075    2076#   2084    2088#
                        2113    2114#   2129    2130#   2141    2142#   2150    2151#   2166    2167#   2180    2181#   2187
                        2188#   2203    2204#   2217    2218#   2224    2225#   2237    2238#   2243    2244#   2249    2250#
                        2256    2257#   2266    2267#   2273    2274#   2281    2282#   2289    2293#   2342    2346#   2358
                        2359#   2370    2371#   2382    2383#   2394    2395#   2417    2421#   2445    2449#   2480    2484#
                        2511    2515#   2549    2553#   2590    2594#   2641    2645#   2682    2686#   2725    2729#   2775
                        2779#   2799    2803#   2829    2833#   2892    2896#   3048    3052#   3110    3114#   3132    3136#
                        3140    3141#   3158    3162#   3202    3206#   3251    3255#   3324    3328#   3366    3370#   3403
                        3407#   3428    3432#   3452    3456#   3486    3490#   3574    3578#   3590    3594#   3621    3625#
                        3664    3668#   3695    3699#   3723    3727#   3783    3787#   3820    3824#   3844    3848#   3877
                        3881#   3916    3920#   3948    3952#   4000    4004#   4106    4110#   4129    4133#   4152    4156#
                        4178    4182#   4200    4204#   4735    4739#
NED   = 010000          1813#   3652    3656
NEDDON  013644          3633    3660#
NEDTST  013466          3626#
NNDD    013726          3659    3661    3663#
NNOOP   004626          2450#
NNOP21  005144          2516#
NOOP    004502          2422#
NOOP21  005006          2485#
NOPAR   011320          3239    3245    3249#
NOWGO   001774          1914    1931    1937#
NXM     006316          2721#
```

```
NXMTSM  006650          2804#
OBUFSV  001102          1744#  1864*  1937*  2424   2439   2456   2471   2487   2502   2522   2537   2597   2603
                        2631   2648   2672   2692   2716   2721   2722*  2736   2766   2781   2787   2836   2883
                        2908   2938   2965   2999   3013   3028   3262   3278   3301   3331   3372   3409   3434
                        3458   3500   3507   3524   3564   3581   3599   3675   3706   3748   3774   3828   3837
                        3851   3885   3927   3955   3966   3978   3984   4142   4165   4190   4239   4272   4682
ONCEE   001164          1673*  1691*  1831#  1863*  1865   1887*  1906   1913*  1920   1938*  3267   3292   3316
                        3319*  3323*  3660   3663*  4926*
OOUT    007302          2849   2889   2891#
OR    = 000200          1811#
OUT     020032          4325#
OUTBUF  024550          1864   1937   2295   2297   2311   2325   2423*  2455*  2486*  2521*  2596*  2647*  2677
                        2691*  2722   2735*  3055   3061   3079   3222*  3261*  3269*  3271*  3274*  3276*  3290
                        3321*  3330*  3499*  3506*  3549   3747*  3827*  3850*  3884*  3954*  3965*  4238*  4710*
                        4720   4725   4745*  4746   4816*  4817   4835*  4836   4842   5007#
PART    011112          3207#
PARTST  010520          3115#
PATST   014514          3788#
PATTST  015154          3882#
PC    =%000007          1711#  2610*  2654*  2698*  2745*  2785*  2810*  3337*  3381*  3415*  3440*  3584*  3677*
                        3708*  3929*  4144*  4192*  4447*  4624*  4927*  4937*
PCNT    001004          1735#  4329*  4330*  4341
PGE   = 002000          1812#  3358
PGETST  011676          3329#
PGTRAP  016624          4113   4123#
PIP   = 020000          1817#
PRTP    011300          3208   3243#
PRTP1   011302          3219   3244#
PS    = 177776          1699#  1700   1945*  2009*  4115*  4139*  4140   4161*  4162   4184*  4252*  4317*  4323*
                        4680*
PSW   = 177776          1700#
PTDONE  024246          4864   4900   4904   4922   4926#
QES     016544          4111#
QESDON  016640          4121   4127#
QQ    = 000001          1869#  1941#  2004#  2054#  2075#  2084#  2113#  2129#  2141#  2150#  2166#  2180#  2187#
                        2203#  2217#  2224#  2237#  2243#  2249#  2256#  2266#  2273#  2281#  2289#  2342#  2358#
                        2370#  2382#  2394#  2417#  2445#  2480#  2511#  2549#  2590#  2641#  2682#  2725#  2775#
                        2799#  2829#  2892#  3048#  3110#  3132#  3140#  3158#  3202#  3251#  3324#  3366#  3403#
                        3428#  3452#  3486#  3574#  3590#  3621#  3664#  3695#  3723#  3783#  3820#  3844#  3877#
                        3916#  3948#  4000#  4106#  4129#  4152#  4178#  4200#  4735#
RANDOM  024312          2296   4940#
RAND1   024326          4942#  4970
RANEND  024464          4976#
RANNU   001154          1827#
RANTS   003714          2294#
RDLIN = 104412          4556   4631#
RDOCT = 104410          4630#  4702   4705   4709
RDTST   005760          2646#
RMRT1   012076          3371#
RMRT2   012254          3408#
RMRT3   012364          3433#
RMRT4   012474          3457#
RSAS    001122          1755#  1881   2001   2442   2452   2474   2505   2518   2540   2583   2872   2874   2876*
                        2877   2934   3106   3342   3466*  3648   3730   3739*  3740   3777   3801   4297   4313*
                        4670*  4671   4882
RSBA    001112          1751#  1948*  1971   2012*  2029   2116*  2117   2120*  2121   2124*  2125   2134*  2135
```

```
                    2145* 2146* 2147  2330  2335  2384* 2386  2389* 2390  2424* 2439  2456* 2471
                    2487* 2502  2522* 2537  2597* 2603  2630  2648* 2671  2692* 2715  2736* 2765
                    2781* 2788  2807* 2836* 2883  2908* 2938  2965* 2999  3015  3033  3056* 3060
                    3080* 3086  3223* 3262* 3278* 3300  3331* 3372* 3409* 3434* 3458* 3500* 3507*
                    3524* 3555  3564* 3581* 3599* 3675* 3706* 3748* 3774  3828* 3837  3851* 3885*
                    3927* 3955* 3966* 3984* 4026* 4042* 4066* 4082* 4142* 4165* 4190* 4239* 4271
                    4642* 4643  4682* 4720* 4725* 4746* 4817* 4836* 4887
RSBAB   001152      1767# 2385*
RSBAE   001134      1760# 2808*
RSCS1   001104      1748# 1947* 1966  2011* 2024  2063* 2064  2067* 2068  2071* 2072  2078* 2079*
                    2080  2348* 2350  2353* 2354  2426* 2427  2458* 2459  2489* 2490  2508  2524*
                    2525  2546  2561* 2580  2599* 2606* 2607  2615  2650* 2651  2659  2694* 2695
                    2706  2747  2784* 2809* 2818  2821* 2845* 2861  2910* 2913  2916  2951  2969*
                    2972  2975  3040  3057* 3058  3081* 3082  3099* 3100  3225* 3226  3236  3264*
                    3265  3281* 3282  3297  3313  3333* 3334  3336* 3339  3357* 3363  3379* 3392
                    3399  3411* 3412  3423  3436* 3437  3439* 3448  3465* 3479  3483  3502* 3503
                    3514* 3521  3526* 3527  3535  3565* 3566  3583* 3601* 3602  3611  3645  3676*
                    3685  3707* 3716  3736  3744  3750* 3751  3754  3756  3807  3811  3831* 3832
                    3834  3853* 3856  3858  3874  3887* 3888  3891  3893  3910  3928* 3937  3961*
                    3962  3969* 3970  3990* 3991  4027* 4030  4032  4036  4044* 4046  4048  4052
                    4068* 4069  4071  4074  4083* 4084  4086  4090  4116* 4124  4143* 4166* 4173
                    4191* 4195  4243* 4249* 4255  4285* 4290* 4308  4314* 4683* 4684  4686  4721*
                    4722  4726* 4728  4729  4749* 4750  4820* 4821  4823  4839* 4840  4867  4930
RSCS1B  001144      1764# 2349* 3651*
RSCS2   001106      1749# 1875* 1879* 1890* 1946* 1953* 1958  1961* 2010* 2017* 2019* 2020  2023*
                    2090  2093* 2094  2096  2099* 2100  2103* 2104  2107* 2108* 2109  2112* 2360*
                    2361* 2365  2401  2562  2626  2667  2702  2761  2783* 2796  2805* 2814  2822
                    2926  2993  3007  3022  3067  3092  3230  3284  3309  3347  3359  3501* 3519*
                    3520* 3538  3635* 3640  3652  3656  3759  3789* 3791  3814  3826* 3830* 3849*
                    3855* 3861  3883* 3890* 3896  3960* 3968* 3972* 3995  4246* 4248* 4254* 4267
                    4304* 4665* 4666  4713* 4851* 4852* 4875
RSCS2B  001146      1765# 2362*
RSCS3   001136      1761#
RSDA    001114      1752# 1949* 1977  2013* 2035  2191* 2192  2195* 2196  2199* 2200  2209* 2210
                    2219* 2220* 2221  2316  2320  2436  2468  2499  2534  2558* 2569  2612  2656
                    2709  2750  2835* 2844* 2944  2985  3177* 3179* 3180  3192  3280* 3380* 3386
                    3505* 3582* 3586  3600* 3636* 3655  3674* 3705* 3771  3922* 3964* 3983* 4041*
                    4081* 4188* 4241* 4279  4282  4647* 4648  4716* 4747* 4754* 4758* 4767* 4776*
                    4785* 4794* 4803* 4812* 4818* 4837* 4892
RSDB    001126      1757# 2015* 2396* 2397* 2408  2413  4914
RSDS    001116      1753# 1962  2462  2528  2621  2756  2846  2850  2856  2858  2880  2923  2982
                    3118  3125  3389  3420  3445  3468  3476  3532  3608  3682  3689  3713  3720
                    3765  3867  3903  3934  3941  4305  4652* 4653  4755  4759  4764  4768  4773
                    4777  4782  4786  4791  4795  4800  4804  4809  4813  4825  4832  4847  4902
RSDT    001132      1759# 1891  1895  1899  1901  4909
RSER    001120      1754# 1876* 1950* 1983  2014* 2041  2228* 2229  2232* 2233  2239* 2240  2245*
                    2246  2251* 2252* 2253  2430  2451* 2477  2493  2517* 2543  2559* 2572  2636
                    2792  2920  2948  2979  3043  3071  3075  3096  3103  3117* 3122  3354  3383
                    3396  3414* 3417  3442  3473  3529  3560  3605  3615  3637  3679  3692  3710
                    3729* 3780  3798  3870  3906  3913  3931  4657* 4658  4828  4871
RSLA    001124      1756# 3137  3145  3149  3186  4662  4924
RSMR    001130      1758# 1951* 1989  2018* 2046  2260* 2261  2268* 2269* 2270  2275* 2276  2283*
                    2284  2560* 2575  2839  3167  3171  3184  3790  3797* 4675* 4676  4919
RSMRR   021650      1687  4675* 4677
RSREG   023544      4447  4856#
RSVCPS  001142      1763# 4114* 4138* 4160* 4187* 4237* 4319  4320*
```

```
RSVEC   001140          1762# 4113* 4137* 4159* 4185* 4236* 4288* 4319*
RSWC    001110          1750# 1952* 1995  2016* 2051  2154* 2155  2158*  2159  2162* 2163  2172* 2173
                        2182* 2183* 2184  2302  2306  2372* 2374  2377*  2378  2425* 2433  2457* 2465
                        2488* 2496  2523* 2531  2557* 2586  2598* 2600   2618  2649* 2662  2693* 2712
                        2737* 2753  2782* 2806* 2837* 2886  2909* 2941   2966* 3002  3018  3036  3054*
                        3078* 3224* 3263* 3279* 3332* 3351  3373* 3410*  3435* 3459* 3498* 3508* 3525*
                        3545  3563* 3580* 3598* 3670* 3673* 3701* 3704*  3749* 3768  3829* 3840  3852*
                        3886* 3923* 3926* 3956* 3959* 3967* 3987* 3989*  4025* 4043* 4067* 4080* 4141*
                        4164* 4189* 4240* 4262  4637* 4638  4681* 4717*  4748* 4819* 4838* 4897
RSWCB   001150          1766# 2373*
RSWCWT  006210          2698#
RSC4DT  001166          1832# 1893* 1897* 1903* 3495  3550  3671  3702   3924  3957  3973  3985  4277
RW      = 000006        1780# 4018  4019  4020  4021
R0      =%000000        1704# 4334* 4516  4535* 4942* 4946* 4951* 4955*  4961* 4963  4964
R1      =%000001        1705# 4874* 1877* 2295* 2740* 2741* 2743* 2744*  3213* 3214* 3215  3978* 3979*
                        3980* 4517  4527* 4528* 4534* 4553  4557* 4568*  4570* 4572* 4574  4577* 4964*
R2      =%000002        1706# 3210* 3211  3212* 3216  3218* 3220* 3228*  3244  3246* 3509* 3510  3515
                        4023* 4024* 4040* 4056  4059  4064* 4065* 4078*  4094  4097  4215* 4232* 4242*
                        4248  4294* 4299* 4304  4518  4533* 4554  4559*  4561  4563  4565* 4566* 4567*
                        4569* 4571* 4576* 4945* 4948* 4954* 4957* 4959*  4960*
R3      =%000003        1707# 2299* 2300* 2313* 2314* 2327* 2328* 2738*  2744  2905* 2906  2961* 2962
                        3030  3374* 3375  3460* 3461  3510* 3511* 3512   3515* 3516* 3517  4213* 4233*
                        4234  4244  4250* 4519  4532* 4555  4558* 4575*  4944* 4949*
R4      =%000004        1708# 2297* 2303  2304  2307  2309  2311* 2317   2318  2321  2323  2325* 2331
                        2332* 2333  2336  2339  2731* 2732  2734* 2741   3375* 3376* 3377  3461* 3462*
                        3463  4214* 4219  4221* 4232  4354  4364* 4368*  4382* 4471  4474* 4494  4500*
                        4520  4531* 4943* 4947* 4952* 4953* 4956* 4958*  4962* 4967  4968
R5      =%000005        1709# 2296* 2302* 2303* 2304  2316* 2317* 2318   2330* 2331* 2333  2739* 2743
                        4227* 4228* 4229* 4246  4355  4356* 4357  4359*  4360  4362  4370  4378* 4379*
                        4380  4381* 4472  4473* 4480* 4482* 4484* 4499*  4521  4530* 4590  4591* 4592
                        4609* 4971*
SAVBAD  001100          1743#
SAVEE   001176          1838# 4234* 4311*
SC      = 100000        1809#
SCOPE   = 104400        1872  1944  2007  2057  2075  2087  2113  2129   2141  2150  2166  2180  2187
                        2203  2217  2224  2237  2243  2249  2256  2266   2273  2281  2292  2345  2358
                        2370  2382  2394  2420  2448  2483  2514  2552   2593  2644  2685  2728  2778
                        2802  2832  2895  3051  3113  3135  3140  3161   3205  3254  3327  3369  3406
                        3431  3455  3489  3577  3593  3624  3667  3698   3726  3786  3823  3847  3880
                        3919  3951  4003  4109  4132  4155  4181  4203   4321  4328  4626# 4738
SECT    015474          3953#
SEEC    024100          4878  4905#
SETPAT  014666          3825#
SHIFT   024344          4946# 4950
SP      =%000006        1710# 1854* 4103* 4112* 4134* 4157* 4183* 4289*  4315* 4318* 4522  4526* 4699*
                        4936*
SRAS    021632          1686  4670# 4672
SRCS2   021614          1685  4665# 4667
SRS8A   021514          1680  4642# 4644
SRSCS1  021666          1688  4680# 4685  4688
SRSDA   021532          1681  4647# 4649
SRSDS   021550          1682  4652# 4654
SRSER   021566          1683  4657# 4659
SRSLA   021604          1684  4662# 4663
SRSWC   021476          1679  4637# 4639
SR0     = 177572        1771# 4010  4022* 4100*
```

```
STRT    000230          1667    1673#
STTEST  001434          1885    1888#   1918
SWR   = 177570          1701#   1702    1884    1904    1924    4331    4391    4393    4395    4406    4434    4438    4448
                        4451    4637    4642    4647    4652    4657    4665    4670    4675    4683    4711    4714    4718
                        4732    4743
SWRCHG  022232          4724    4728#
SW10  = 002000          1657#   4331    4434
SW11  = 004000          1656#   4406
SW12  = 010000          1655#   4332    4334
SW13  = 020000          1654#   4438
SW14  = 040000          1653#   4395
SW15  = 100000          1652#
SW8   = 000400          1660#   4391
SW9   = 001000          1659#   4451
TIMES   020466          1939    1940*   2062*   2902    2903*   3115*   3163    3164*   3207*   4006    4007*   4135*   4210
                        4211*   4322*   4410    4425#
TIMSV   001170          1833#   1939*   2062    2902*   3115    3163*   3207    4006*   4135    4210*   4322
TKS     022144          4716#
TKSEL   021736          1867    4699#
TK1     022112          4711#   4733
TK2     022120          4712#   4734
TMEOUT  011106          3183    3199#
TPB     001020          1740#   4365*   4370*
TPS     001016          1739#   4366    4371
TRE   = 040000          1808#   2821    3357    4285
TRY     001330          1876#   1880
TRYNX   001606          1889    1902    1915#   4325
TSTAGN  011140          3211#   3247
TSTNEM  006716          2812#
TSTVEC  017442          4236    4254#
TST1    001316          1872#
TST10   003022          2129#
TST100  015000          3847#
TST101  015152          3880#
TST102  015346          3919#
TST103  015472          3951#
TST104  015776          4003#
TST105  016542          4109#
TST106  016640          4132#
TST107  016744          4155#
TST11   003054          2141#
TST110  017062          4181#
TST111  017170          4203#
TST112  022262          4738#
TST12   003100          2150#
TST13   003162          2166#
TST14   003214          2180#
TST15   003240          2187#
TST16   003322          2203#
TST17   003354          2217#
TST2    002024          1944#
TST20   003400          2224#
TST21   003442          2237#
TST22   003464          2243#
TST23   003506          2249#
TST24   003532          2256#
```

TST25	003570	2266#											
TST26	003616	2273#											
TST27	003654	2281#											
TST3	002246	2007#											
TST30	003712	2292#											
TST31	004132	2345#											
TST32	004204	2358#											
TST33	004250	2370#											
TST34	004320	2382#											
TST35	004372	2394#											
TST36	004500	2420#											
TST37	004624	2448#											
TST4	002460	2057#											
TST40	005004	2483#											
TST41	005142	2514#											
TST42	005334	2552#											
TST43	005530	2593#											
TST44	005756	2644#											
TST45	006144	2685#											
TST46	006332	2728#											
TST47	006542	2778#											
TST5	002552	2075#											
TST50	006646	2802#											
TST51	007002	2832#											
TST52	007302	2895#											
TST53	010220	3051#											
TST54	010516	3113#											
TST55	010574	3135#											
TST56	010610	3140#											
TST57	010720	3161#											
TST6	002600	2087#											
TST60	011110	3205#											
TST61	011332	3254#											
TST62	011674	3327#											
TST63	012074	3369#											
TST64	012252	3406#											
TST65	012362	3431#											
TST66	012472	3455#											
TST67	012642	3489#											
TST7	002740	2113#											
TST70	013274	3577#											
TST71	013350	3593#											
TST72	013464	3624#											
TST73	013734	3667#											
TST74	014072	3698#											
TST75	014216	3726#											
TST76	014512	3786#											
TST77	014664	3823#											
TTAGG	002250	2009#											
TTMOUT	007300	2855	2890#										
TTVEC	017644	4288	4294#										
TYPE =	104402	1886	1908	1912	1928	1929	3155	3662	4333	4436	4440	4446	4498	4538
		4581	4601	4604	4608	4627#	4701	4704	4708	4741	4752	4762	4771	4780
		4789	4798	4807	4845	4858	4861	4866	4870	4874	4881	4886	4891	4896
		4901	4908	4913	4918	4923								
TYPE0 =	104404	4445	4628#	4860	4863	4868	4872	4876	4883	4888	4893	4898	4903	4910

```
                4915    4920    4925
TYPES = 104406  1911    1927    4629#
UNCMP   001162  1830#   1882*   1888    1915*   1922*   2442    2452    2474    2505    2518    2540    2583    2932
                3731    3739    3777    3802
UNITST  006234  2730#
UNITSV  001160  1829#   1881*   1888    3629    4217
UNNUM   001156  1828#   1883*   1890    1910    1917*   1923*   1926    1946    1961    2010    2023    2112    2360
                2363    2400    2564    2624    2665    2700    2732    2739    2759    2812    2823    2864    2927
                2994    3008    3023    3065    3090    3232    3285    3307    3345    3520    3539    3761    3793
                3815    3863    3898    3993    4219    4227    4254    4265    4703*   4743*   4852
UP      = 000000  1781#   4018    4019    4020    4021
WAITRY  024256  2610    2654    2698    2745    2785    2810    3337    3381    3415    3440    3584    3677    3708
                3929    4144    4192    4929#
WC      = 000010  1796#   1997    2053    2157    2161    2165    2186    2376    2380    2435    2467    2498    2533
                2588    2602    2620    2664    2714    2755    2888    2943    3004    3020    3038    3353    3770
                3842    4264    4731    4894
WCE     = 040000  1814#
WCEDNE  011644  3277    3319#
WCEDON  011666  3272    3323#
WCETST  011334  3260#
WCETT   011342  3262#   3322
WORK    001200  1850#   2261*   2262*   2263    2276*   2277*   2278    2284*   2285*   2286    2294*   2398*   2404*
                2575*   2576*   2577    2838*   2841*   2864*   2868    2870*   3013*   3014*   3015    3028*   3029*
                3032*   3033    3055*   3056    3079*   3080    3144*   3147*   3178*   3182*   3213    3223    3229*
                3467*   3470*   3627*   3629    3632*   3655*   3975*   3977*   3981*   4117*   4118*   4167*   4168*
                4216*   4217    4223*   4296*   4297    4301*   4313    4706*   4707*   4717    4929*   4932*   4933
                4965*   4969*
WORK1   001202  1851#   2967*   2968*   2988    2990    3188*   3190*   4212*   4225    4231*   4286    4714*   4715*
                4716
WORK2   001204  1852#   3494*   3497*   3498    3525    4711*   4732
WRCKT   006146  2690#
WRTLCK  022264  1670    4740#
WRTST   005532  2595#
WT      022204  4722#   4727
WTDV    017440  4253#
WTDV1   017762  4287    4312    4317#
WTDV2   017642  4289    4292#   4315
WTE     022214  4719    4725#
WWP     = 000004  1835#   3212
ZERONE  004374  2395#
$ENDAD  020102  1663    4337#
$END1   020112  4335    4339#
.       = 026150  1661#   1662#   1664#   1666#   1669#   1672#   1678#   1690#   1731#   1742#   1866    1886    1908
                1929    1930#   1959    1963    1967    1972    1978    1984    1990    1996    2002    2021    2025
                2030    2036    2042    2047    2052    2065    2069    2073    2081    2091    2101    2105    2110
                2118    2122    2126    2137    2148    2156    2160    2164    2175    2185    2193    2197    2201
                2212    2222    2230    2234    2241    2247    2254    2264    2271    2279    2287    2351    2355
                2367    2375    2379    2387    2391    2410    2415    2428    2431    2434    2437    2440    2443
                2453    2460    2463    2466    2469    2472    2475    2478    2491    2494    2497    2500    2503
                2506    2509    2519    2526    2529    2532    2535    2538    2541    2544    2547    2567    2570
                2573    2578    2581    2584    2587    2601    2604    2608    2613    2616    2619    2622    2628
                2634    2637    2652    2657    2660    2663    2669    2675    2680    2696    2704    2707    2710
                2713    2719    2748    2751    2754    2757    2763    2769    2790    2793    2797    2816    2819
                2826    2851    2859    2862    2878    2881    2884    2887    2918    2921    2924    2930    2936
                2939    2942    2945    2949    2952    2977    2980    2983    2986    2997    3000    3003    3011
                3016    3019    3026    3034    3037    3041    3044    3063    3069    3072    3076    3088    3094
```

		3097	3101	3104	3107	3119	3123	3126	3138	3155	3170	3172	3195	3234
		3237	3298	3304	3311	3314	3340	3343	3349	3352	3355	3361	3364	3384
		3387	3390	3393	3397	3400	3418	3421	3424	3443	3446	3449	3474	3477
		3480	3484	3530	3533	3536	3542	3547	3557	3561	3587	3603	3606	3609
		3612	3616	3638	3643	3646	3649	3653	3657	3662	3680	3683	3686	3690
		3693	3711	3714	3717	3721	3734	3737	3741	3745	3752	3757	3763	3766
		3769	3772	3775	3778	3781	3795	3799	3805	3808	3812	3818	3835	3838
		3841	3859	3865	3868	3871	3875	3894	3901	3904	3907	3911	3914	3932
		3935	3938	3942	3997	4031	4047	4070	4085	4125	4169	4174	4196	4263
		4269	4275	4283	4306	4309	4333	4334#	4341	4343#	4367	4372	4425	4440
		4441#	4446	4447#	4449	4457	4458	4467	4487	4490	4495	4505#	4538	4542
		4581	4595	4601	4612#	4701	4702#	4704	4708	4723	4741	4742#	4752	4753#
		4756	4760	4762	4765	4769	4771	4774	4778	4780	4783	4787	4789	4792
		4796	4798	4801	4805	4807	4810	4814	4826	4829	4833	4843	4845	4846#
		4848	4858	4861	4862#	4866	4870	4871#	4874	4881	4882#	4886	4887#	4891
		4892#	4896	4897#	4901	4902#	4908	4909#	4913	4914#	4918	4919#	4923	4924#
		5007#	5008#											
.CLRDK	023526	4632	4851#											
.HLT	020470	1857	4434#											
.HLTCT	020624	4443*	4457#	4856	4877	4879	4884	4889	4894	4899	4905*	4906	4911	4916
		4921												
.ILLUP	021164	4514	4541#											
.KIT	020440	4396	4405	4411	4418#	4455								
.OVER	020444	4394	4419#											
.POWER	021040	1855	4514#	4536										
.POWUP	021104	4523	4526#											
.PR	021014	4464*	4466*	4468*	4469*	4474	4478	4488*	4489	4492*	4494	4498	4505#	
.PRF	020722	4476	4479	4484#										
.PRL	020700	4477#	4493											
.PTIT	020660	4465	4470#											
.PUVEC	021172	4514*	4515*	4523*	4545#									
.QQ. =	000016	4613#	4626	4627#	4628#	4629#	4630#	4631#	4632#	4633#				
.RDLIN	021314	4590#	4631											
.RDOCT	021176	4551#	4630											
.SAVR6	021170	4522*	4526	4544#										
.SCOPE	020276	4391#	4626											
.SVLAD	020420	4402	4407	4413#	4421									
.SWOPT=	167400	1645#	1652	1653	1654	1655	1656	1657	1658	1659	1660	1661	4331	4332
		4334	4391	4434	4438	4451								
.TBIT	020116	4341#												
.TRAP	021440	1859	4620#											
.TRP	021456	4623	4624#											
.TYPE	020140	4354#	4627											
.TYPEB	020626	4464#												
.TYPEO	020636	4466#	4628											
.TYPES	020646	4468#	4629											

```
END          1#    1645#
LOPDR     1783#    4018    4019    4020    4021
NEWTST       1#    1645#    1869    1941    2004    2054    2075    2084    2113    2129    2141    2150    2166    2180    2187
          2203     2217    2224    2237    2243    2249    2256    2266    2273    2281    2289    2342    2358    2370    2382
          2394     2417    2445    2480    2511    2549    2590    2641    2682    2725    2775    2799    2829    2892    3048
          3110     3132    3140    3158    3202    3251    3324    3366    3403    3428    3452    3486    3574    3590    3621
          3664     3695    3723    3783    3820    3844    3877    3916    3948    4000    4106    4129    4152    4178    4200
          4735
ODT11        1#
ODT11X       1#
POP          1#    1728#    4401    4404    4499    4530    4575
PRINT        1#    1728#    1886    1908    1929    3155    3662    4333    4440    4446    4538    4580    4600    4701    4704
          4708     4741    4752    4762    4771    4780    4789    4798    4807    4845    4858    4861    4866    4870    4874
          4881     4886    4891    4896    4901    4908    4913    4918    4923
PUSH         1#    1728#    4398    4470    4516    4553
SCOPE.       1#    1869#    1941#    2004#    2054#    2075#    2084#    2113#    2129#    2141#    2150#    2166#    2180#    2187#    2203#
          2217#    2224#    2237#    2243#    2249#    2256#    2266#    2273#    2281#    2289#    2342#    2358#    2370#    2382#    2394#
          2417#    2445#    2480#    2511#    2549#    2590#    2641#    2682#    2725#    2775#    2799#    2829#    2892#    3048#    3110#
          3132#    3140#    3158#    3202#    3251#    3324#    3366#    3403#    3428#    3452#    3486#    3574#    3590#    3621#    3664#
          3695#    3723#    3783#    3820#    3844#    3877#    3916#    3948#    4000#    4106#    4129#    4152#    4178#    4200#    4735#
SET       4613#    4626    4627    4628    4629    4630    4631    4632
TYPADR       1#
TYPBIT       1#    1728#
TYPCHR       1#    1728#
TYPDEC       1#    1728#
TYPLIN       1#    1728#
TYPOCS       1#    1728#    1909    1926
TYPOCT       1#    1728#    4444    4859    4862    4867    4871    4875    4882    4887    4892    4897    4902    4909    4914
          4919     4924
TYPTXT       1#    1728#    1886    1908    1929    3154    3662    4701    4704    4708    4740    4752    4762    4771    4780
          4789     4798    4807    4845    4858    4861    4865    4869    4873    4881    4886    4891    4896    4901    4908
          4913     4918    4923
$CATCH       1#    1645#    1661
$CMTAG       1#    1645#    1731
$DONE        1#    1645#    4326
$EQUAT       1#    1645#    1697
$HLT         1#    1645#    4426
$KMMR        1#
$OCTAL       1#    1645#    4458
$POWER       1#    1645#    4506
$RAND2       1#
$RAND4       1#
$RDDEC       1#
$RDLIN       1#    1645#    4583
$RDOCT       1#    1645#    4546
$SCOPE       1#    1645#    4384
$SET      4613#    4626    4627    4628    4629    4630    4631    4632
$SETUP       1#    1645#    1854
$SMMR        1#
$SWDOC       1#    1645#    1649
$TRAP        1#    1645#    4613
$TYPE        1#    1645#    4344
$TYPEA       1#
$TYPED       1#
$UMMR        1#
```

```
ADC     4330  4952  4954  4956  4957  4959  4962
ADD     2632  2673  2717  2767  3014  3029  3246  3302  3552  3554  4273  4329  4375  4378  4623
        4936  4951  4953  4955  4958  4960  4961
ASL     1915  4946
BCC     2140  2178  2215  3272  4224  4302
BCS     1916  3277  3633
BEQ     1866  1878  1889  1900  1907  1959  1963  1967  1972  1978  1984  1990  1996  2002  2021
        2025  2030  2036  2042  2047  2052  2065  2069  2073  2081  2091  2095  2101  2105  2110
        2118  2122  2126  2137  2148  2156  2160  2164  2175  2185  2193  2197  2201  2212  2222
        2230  2234  2241  2247  2254  2264  2271  2279  2287  2301  2305  2315  2319  2329  2334
        2341  2355  2367  2375  2379  2387  2391  2403  2410  2415  2428  2431  2434  2437  2440
        2443  2460  2463  2466  2469  2472  2478  2491  2494  2497  2500  2503  2506  2509  2526
        2529  2532  2535  2538  2544  2547  2567  2570  2573  2578  2581  2584  2587  2601  2604
        2613  2616  2619  2622  2628  2634  2637  2657  2660  2663  2669  2675  2680  2704  2707
        2710  2713  2719  2748  2751  2754  2757  2763  2769  2790  2793  2797  2816  2819  2826
        2847  2855  2857  2859  2862  2869  2873  2878  2881  2884  2887  2907  2918  2921  2924
        2930  2936  2939  2942  2945  2949  2952  2963  2977  2980  2983  2986  2989  2991  2997
        3000  3003  3011  3016  3019  3026  3034  3037  3041  3044  3063  3069  3072  3076  3088
        3094  3097  3101  3104  3107  3119  3123  3126  3138  3148  3152  3168  3172  3181  3183
        3185  3195  3234  3237  3245  3268  3288  3293  3238  3304  3311  3314  3317  3340  3343
        3349  3355  3361  3364  3384  3390  3393  3397  3400  3418  3421  3424  3443  3446  3449
        3474  3477  3480  3484  3530  3533  3536  3542  3547  3557  3561  3587  3606  3609  3612
        3616  3630  3638  3643  3646  3649  3653  3680  3683  3690  3693  3711  3714  3721  3734
        3737  3741  3745  3752  3757  3763  3766  3769  3772  3775  3781  3795  3799  3805  3808
        3812  3818  3835  3838  3841  3859  3865  3868  3871  3875  3894  3901  3904  3907  3911
        3914  3925  3932  3935  3942  3974  3986  3992  3997  4037  4053  4057  4075  4091  4095
        4125  4174  4196  4218  4259  4263  4269  4275  4280  4283  4287  4306  4309  4312  4335
        4361  4392  4394  4409  4421  4435  4487  4490  4560  4593  4685  4719  4756  4760  4765
        4769  4774  4778  4783  4787  4792  4796  4801  4805  4810  4814  4826  4829  4833  4843
        4848  4880  4885  4890  4895  4900  4907  4912  4917  4922  4934  4966
BGT     4411  4562
BIC     1673  1863  1887  1913  2262  2277  2285  2332  2563  2576  2911  2933  2968  2970  3146
        3150  3187  3193  3323  3358  3376  3462  3511  3516  3732  3803  3899  3972  4229  4256
        4379  4715  4905
BICB    4597
BIS     1691  1903  1938  2361  2364  2400  2565  2606  2625  2666  2701  2760  2783  2805  2813
        2824  2915  2928  2974  2995  3009  3024  3066  3091  3232  3286  3308  3319  3346  3357
        3501  3540  3641  3663  3761  3789  3793  3816  3826  3830  3849  3855  3863  3883  3890
        3898  3960  3968  3994  4231  4266  4713  4926
BISB    4491
BIT     1865  1884  1888  1904  1906  1920  1924  1989  2024  2046  2270  2427  2442  2452  2459
        2474  2490  2505  2518  2525  2540  2583  2796  2839  2846  2850  2856  3137  3167  3171
        3184  3267  3292  3316  3468  3629  3652  3656  3660  3751  3777  3991  4217  4225  4286
        4297  4331  4357  4391  4395  4406  4434  4438  4451  4478  4684  4718  4877  4879  4884
        4889  4894  4899  4906  4911  4916  4921
BLT     4564
BMI     3217  3335  3413  3438  3522  3603  3686  3854  3889  4931
BNE     1885  1892  1896  1902  1905  1921  1925  2405  2453  2475  2519  2541  2733  2840  2842
        2851  3031  3170  3191  3352  3378  3387  3464  3469  3471  3496  3513  3518  3551  3657
        3661  3672  3703  3778  3958  3982  4119  4169  4220  4226  4245  4251  4278  4298  4332
        4358  4363  4369  4377  4396  4407  4439  4452  4479  4493  4495  4529  4599  4606  4733
        4857  4878  4950  4970
BPL     2608  2652  2696  2914  2973  3059  3083  3227  3266  3283  3504  3528  3567  3717  3755
        3833  3857  3892  3938  3963  3971  4031  4033  4047  4049  4070  4072  4085  4087  4367
        4372  4449  4595  4687  4723  4730  4751  4822  4824  4841
BR      1880  1894  1898  1914  1918  1931  2310  2324  2338  2340  2849  2871  2889  3005  3153
```

```
        3197  3219  3273  3318  3553  3604  3634  3659  3976  3988  4035  4039  4051  4055  4089
        4101  4146  4172  4194  4230  4235  4247  4373  4397  4402  4405  4465  4467  4476  4542
        4573  4582  4602  4639  4644  4649  4654  4659  4663  4667  4672  4677  4688  4724  4727
        4734  4935
CLC     2133  2171  2208  3270  4222  4295  4300
CLR     1861  1862  1875  1893  1897  1923  2079  2108  2146  2183  2220  2252  2269  2396  2407
        2647  2731  2838  2844  2865  3218  3228  3229  3250  3261  3274  3280  3291  3467  3505
        3506  3628  3636  3964  3980  3983  4015  4040  4041  4078  4081  4100  4105  4167  4212
        4213  4214  4238  4294  4320  4468  4527  4557  4745  4835  4929  4945
CLRB    2362  4453  4475  4477  4497  4607
CMF     1891  1895  1899  1901  1958  1962  1966  1995  2020  2051  2064  2068  2072  2080  2090
        2094  2100  2104  2109  2117  2121  2125  2136  2155  2159  2163  2174  2192  2196  2200
        2211  2229  2233  2240  2246  2263  2278  2286  2304  2318  2333  2350  2354  2366  2374
        2378  2386  2390  2402  2409  2414  2433  2439  2462  2465  2471  2477  2496  2502  2508
        2528  2531  2537  2543  2546  2566  2577  2580  2586  2600  2603  2612  2615  2621  2627
        2633  2656  2659  2668  2674  2679  2703  2706  2709  2718  2732  2747  2750  2756  2762
        2768  2789  2815  2818  2825  2858  2861  2872  2880  2883  2886  2917  2920  2923  2929
        2935  2938  2941  2951  2976  2979  2982  2988  2990  2996  2999  3002  3010  3015  3018
        3025  3030  3033  3036  3040  3062  3068  3071  3087  3093  3096  3100  3118  3125  3151
        3180  3194  3233  3236  3243  3244  3287  3297  3303  3310  3313  3339  3348  3360  3363
        3377  3383  3386  3389  3392  3399  3417  3420  3423  3442  3445  3448  3463  3476  3479
        3483  3512  3517  3529  3532  3535  3541  3546  3556  3586  3605  3608  3611  3642  3645
        3679  3682  3689  3713  3720  3733  3736  3744  3756  3762  3765  3768  3774  3780  3794
        3798  3804  3807  3811  3817  3834  3837  3840  3858  3864  3867  3874  3893  3900  3903
        3906  3910  3934  3941  3996  4036  4052  4056  4074  4090  4094  4123  4124  4173  4195
        4198  4219  4258  4268  4274  4279  4282  4305  4308  4376  4403  4494  4592  4732  4755
        4759  4764  4768  4773  4777  4782  4786  4791  4795  4800  4804  4809  4813  4825  4828
        4832  4847
CMPB    4362  4393  4410  4561  4563  4598  4605
COM     4707
DEC     1877  2300  2314  2328  2404  2841  2854  2870  3147  3169  3182  3190  3981  4118  4250
        4311  4368  4949  4965  4969
EMT     1698
HALT    1662  1930  4450  4524  4541  4742  4753  4763  4772  4781  4790  4799  4808  4846
INC     1879  1917  2734  3179  3470  3631  4168  4221  4228  4233  4299  4437  4528  4932
INCB    4413  4418  4488  4492
JMP     1667  1670  1674  1679  1680  1681  1682  1683  1684  1685  1686  1687  1688  1692  1867
        1932  2954  2964  3046  3239  3247  3322  4121  4325  4339  4455  4864  4904
JSR     2296  2610  2654  2698  2745  2785  2810  3337  3381  3415  3440  3584  3677  3708  3929
        4144  4192  4337  4447
MOV     1854  1855  1856  1857  1858  1859  1860  1864  1874  1876  1881  1882  1883  1890  1910
        1922  1926  1937  1939  1940  1945  1946  1947  1948  1949  1950  1951  1952  1953  1961
        2009  2010  2011  2012  2013  2014  2015  2016  2017  2018  2019  2023  2062  2063  2067
        2071  2078  2093  2096  2097  2099  2103  2107  2112  2116  2120  2124  2132  2134  2135
        2145  2154  2158  2162  2170  2172  2173  2182  2191  2195  2199  2207  2209  2210  2219
        2228  2239  2245  2251  2260  2261  2268  2275  2276  2283  2284  2294  2295  2297  2298
        2299  2302  2303  2306  2307  2311  2312  2313  2316  2317  2320  2321  2325  2326  2327
        2330  2331  2335  2336  2348  2360  2363  2365  2372  2384  2397  2398  2399  2401  2408
        2412  2413  2423  2424  2425  2426  2451  2455  2456  2457  2458  2486  2487  2488  2489
        2517  2521  2522  2523  2524  2557  2558  2559  2560  2561  2562  2564  2575  2596  2597
        2598  2599  2624  2626  2630  2631  2648  2649  2650  2665  2667  2671  2672  2677  2678
        2691  2692  2693  2694  2700  2702  2715  2716  2721  2722  2735  2736  2737  2738  2739
        2740  2741  2743  2744  2759  2761  2765  2766  2781  2782  2784  2787  2788  2806  2807
        2808  2809  2812  2814  2821  2822  2823  2835  2836  2837  2845  2853  2864  2874  2876
        2902  2903  2905  2906  2908  2909  2910  2912  2916  2926  2927  2932  2934  2961  2962
        2965  2966  2967  2969  2971  2975  2993  2994  3007  3008  3013  3022  3023  3028  3054
```

```
        3055  3056  3057  3060  3061  3065  3067  3078  3079  3080  3081  3084  3086  3090  3092
        3099  3115  3117  3144  3145  3149  3163  3164  3166  3177  3178  3186  3188  3192  3207
        3208  3209  3210  3212  3213  3214  3220  3222  3223  3224  3225  3230  3231  3249  3262
        3263  3264  3269  3278  3279  3281  3284  3285  3290  3294  3300  3301  3307  3309  3321
        3330  3331  3332  3333  3336  3345  3347  3359  3372  3373  3374  3375  3379  3380  3409
        3410  3411  3414  3434  3435  3436  3439  3458  3459  3460  3461  3465  3466  3494  3497
        3498  3499  3500  3502  3507  3508  3509  3510  3514  3515  3519  3520  3524  3525  3526
        3538  3539  3544  3545  3549  3555  3563  3564  3565  3580  3581  3582  3583  3598  3599
        3600  3601  3627  3635  3640  3655  3670  3673  3674  3675  3676  3701  3704  3705  3706
        3707  3729  3730  3731  3739  3747  3748  3749  3750  3759  3760  3791  3792  3797  3801
        3802  3814  3815  3827  3828  3829  3831  3850  3851  3852  3853  3861  3862  3884  3885
        3886  3887  3896  3897  3922  3923  3926  3927  3928  3954  3955  3956  3959  3961  3965
        3966  3967  3969  3975  3977  3978  3979  3984  3987  3989  3990  3993  3995  4006  4007
        4008  4009  4013  4014  4016  4017  4018  4019  4020  4021  4022  4023  4024  4025  4026
        4027  4042  4043  4044  4058  4059  4063  4064  4065  4066  4067  4068  4080  4082  4083
        4096  4097  4103  4104  4112  4113  4114  4115  4116  4117  4134  4135  4137  4138  4139
        4140  4141  4142  4143  4149  4157  4159  4160  4161  4162  4163  4164  4165  4166  4183
        4184  4185  4187  4188  4189  4190  4191  4210  4211  4215  4216  4227  4232  4234  4236
        4237  4239  4240  4241  4242  4243  4246  4248  4249  4252  4254  4255  4257  4265  4267
        4271  4272  4285  4288  4289  4290  4296  4304  4313  4314  4315  4317  4318  4319  4322
        4323  4324  4334  4354  4355  4356  4359  4374  4380  4381  4382  4398  4399  4401  4404
        4414  4415  4419  4422  4441  4444  4464  4471  4472  4473  4474  4499  4500  4501  4502
        4514  4515  4516  4517  4518  4519  4520  4521  4522  4523  4526  4530  4531  4532  4533
        4534  4535  4536  4537  4551  4552  4553  4554  4555  4558  4574  4575  4576  4577  4590
        4591  4609  4620  4622  4624  4637  4638  4642  4643  4647  4648  4652  4653  4657  4658
        4662  4665  4666  4670  4671  4675  4676  4680  4681  4682  4683  4699  4703  4706  4710
        4711  4714  4716  4717  4720  4721  4725  4726  4728  4743  4746  4747  4748  4749  4754
        4758  4767  4776  4785  4794  4803  4812  4816  4817  4818  4819  4820  4836  4837  4838
        4839  4851  4852  4859  4862  4867  4871  4875  4882  4887  4892  4897  4902  4909  4914
        4919  4924  4940  4941  4942  4943  4944  4963  4964  4967  4968
MOVB    2232  2349  2353  2373  2377  2385  2389  3651  4364  4365  4370  4412  4443  4466  4469
        4496  4559  4596  4603
NOP     2341  2742  4061  4102  4700
RESET   1954  2089  4336
ROL     2139  2177  2214  2867  3271  3276  3632  4223  4301  4480  4482  4484  4568  4569  4570
        4571  4572  4947  4948
ROLB    4481  4483  4485
ROR     3189  4565  4566  4567
RTI     4291  4316  4383  4416  4423  4454  4503  4539  4578  4610  4853
RTS     4927  4937  4971
SEC     2866  3275
SUB     3032  4442  4621
TRAP    4623  4626  4627  4628  4629  4630  4631  4632
TST     1971  1977  1983  2001  2029  2035  2041  2147  2184  2221  2253  2309  2323  2339  2430
        2436  2468  2493  2499  2534  2569  2572  2618  2636  2662  2712  2753  2792  2868  2877
        2944  2948  2985  3043  3075  3103  3106  3122  3211  3215  3216  3342  3351  3354  3396
        3473  3495  3550  3560  3615  3637  3648  3671  3685  3692  3702  3710  3716  3740  3771
        3790  3870  3913  3924  3931  3937  3957  3973  3985  4010  4032  4048  4071  4086  4244
        4262  4277  4400  4420  4448  4729  4842  4856  4933
TSTB    2607  2651  2695  2913  2972  3058  3082  3226  3265  3282  3334  3412  3437  3503  3521
        3527  3566  3602  3754  3832  3856  3888  3891  3962  3970  4030  4046  4069  4084  4360
        4366  4371  4408  4486  4489  4594  4686  4722  4750  4821  4823  4840  4930
WAIT    4253  4292
.ASCIZ  1887  1909  1930  3156  3663  4334  4441  4447  4539  4582  4602  4702  4705  4709  4742
        4753  4763  4772  4781  4790  4799  4808  4846  4859  4862  4867  4871  4875  4882  4887
        4892  4897  4902  4909  4914  4919  4924
```

```
.BLKB   4612
.BLKW   4343  4505  5007  5008
.ENABL     1  1645
.END    5009
.ENDC   1728  1855  1861  1872  1944  2007  2057  2075  2087  2113  2129  2141  2150  2166  2180
        2187  2203  2217  2224  2237  2243  2249  2256  2266  2273  2281  2292  2345  2358  2370
        2382  2394  2420  2448  2483  2514  2552  2593  2644  2685  2728  2778  2802  2832  2895
        3051  3113  3135  3140  3161  3205  3254  3327  3369  3406  3431  3455  3489  3577  3593
        3624  3667  3698  3726  3786  3823  3847  3880  3919  3951  4003  4109  4132  4155  4181
        4203  4331  4333  4334  4340  4396  4424  4438  4440  4448  4456  4522  4530  4539  4540
        4593  4613  4627  4628  4629  4630  4631  4632  4633  4738
.EVEN   1887  1909  1930  3156  3663  4334  4441  4447  4539  4582  4602  4702  4705  4709  4742
        4753  4763  4772  4781  4790  4799  4808  4846  4859  4862  4867  4871  4875  4882  4887
        4892  4897  4902  4909  4914  4919  4924
.IF     1712  1854  1861  1869  1941  2004  2054  2075  2084  2113  2129  2141  2150  2166  2180
        2187  2203  2217  2224  2237  2243  2249  2256  2266  2273  2281  2289  2342  2358  2370
        2382  2394  2417  2445  2480  2511  2549  2590  2641  2682  2725  2775  2799  2829  2892
        3048  3110  3132  3140  3158  3202  3251  3324  3366  3403  3428  3452  3486  3574  3590
        3621  3664  3695  3723  3783  3820  3844  3877  3916  3948  4000  4106  4129  4152  4178
        4200  4331  4332  4334  4339  4391  4424  4434  4438  4447  4451  4522  4530  4538  4539
        4592  4612  4626  4627  4628  4629  4630  4631  4632  4735
.IFF    1855  4332  4340  4396  4434  4451  4540  4593  4613  4626  4627  4628  4629  4630  4631
        4632
.IIF    1652  1653  1654  1655  1656  1657  1658  1659  1660  1661  4341  4416  4420  4425  4457
        4458  4612
.IRP    4398  4401  4404  4471  4499  4516  4530  4553  4575  4626
.LIST      1  1645  1662  1728  1869  1873  1887  1909  1930  1941  1945  2004  2008  2054  2058
        2075  2076  2084  2088  2113  2114  2129  2130  2141  2142  2150  2151  2166  2167  2180
        2181  2187  2188  2203  2204  2217  2218  2224  2225  2237  2238  2243  2244  2249  2250
        2256  2257  2266  2267  2273  2274  2281  2282  2289  2293  2342  2346  2358  2359  2370
        2371  2382  2383  2394  2395  2417  2421  2445  2449  2480  2484  2511  2515  2549  2553
        2590  2594  2641  2645  2682  2686  2725  2729  2775  2779  2799  2803  2829  2833  2892
        2896  3048  3052  3110  3114  3132  3136  3140  3141  3156  3158  3162  3202  3206  3251
        3255  3324  3328  3366  3370  3403  3407  3428  3432  3456  3452  3486  3490  3574  3578
        3590  3594  3621  3625  3663  3664  3668  3695  3699  3723  3727  3783  3787  3820  3824
        3844  3848  3877  3881  3916  3920  3948  3952  4000  4004  4106  4110  4129  4133  4152
        4156  4178  4182  4200  4204  4334  4441  4447  4539  4582  4602  4613  4626  4627  4628
        4629  4630  4631  4632  4633  4702  4705  4709  4735  4739  4742  4753  4763  4772  4781
        4790  4799  4808  4846  4859  4862  4867  4871  4875  4882  4887  4892  4897  4902  4909
        4914  4919  4924
.MACR    1783
.MACRO      1  4613
.MCALL   1645  1728  1869  1941  2004  2054  2075  2084  2113  2129  2141  2150  2166  2180  2187
         2203  2217  2224  2237  2243  2249  2256  2266  2273  2281  2289  2342  2358  2370  2382
         2394  2417  2445  2480  2511  2549  2590  2641  2682  2725  2775  2799  2829  2892  3048
         3110  3132  3140  3158  3202  3251  3324  3366  3403  3428  3452  3486  3574  3590  3621
         3664  3695  3723  3783  3820  3844  3877  3916  3948  4000  4106  4129  4152  4178  4200
         4735
.NLIST      1  1645  1662  1728  1869  1873  1887  1909  1930  1941  1945  2004  2008  2054  2058
         2075  2076  2084  2088  2113  2114  2129  2130  2141  2142  2150  2151  2166  2167  2180
         2181  2187  2188  2203  2204  2217  2218  2224  2225  2237  2238  2243  2244  2249  2250
         2256  2257  2266  2267  2273  2274  2281  2282  2289  2293  2342  2346  2358  2359  2370
         2371  2382  2383  2394  2395  2417  2421  2445  2449  2480  2484  2511  2515  2549  2553
         2590  2594  2641  2645  2682  2686  2725  2729  2775  2779  2799  2803  2829  2833  2892
         2896  3048  3052  3110  3114  3132  3136  3140  3141  3156  3158  3162  3202  3206  3251
         3255  3324  3328  3366  3370  3403  3407  3428  3432  3452  3456  3486  3490  3574  3578
```

F12

DERSAA.P11 CROSS REFERENCE TABLE -- PERMANENT SYMBOLS

```
            3590  3594  3621  3625  3663  3664  3668  3695  3699  3723  3727  3783  3787  3820  3824
            3844  3848  3877  3881  3916  3920  3948  3952  4000  4004  4106  4110  4129  4133  4152
            4156  4178  4182  4200  4204  4334  4441  4447  4539  4582  4602  4613  4626  4627  4628
            4629  4630  4631  4632  4633  4702  4705  4709  4735  4739  4742  4753  4763  4772  4781
            4790  4799  4808  4846  4859  4862  4867  4871  4875  4882  4887  4892  4897  4902  4909
            4914  4919  4924
.PAGE       1696  1731  4384  4426  4458  4506  4546  4583
.REM           1   549  1097
.REPT       1662
.SBTTL      1869  1941  2004  2054  2084  2289  2342  2417  2445  2480  2511  2549  2590  2641  2682
            2725  2775  2799  2829  2892  3048  3110  3132  3158  3202  3251  3324  3366  3403  3428
            3452  3486  3574  3590  3621  3664  3695  3723  3783  3820  3844  3877  3916  3948  4000
            4106  4129  4152  4178  4200  4326  4344  4384  4426  4458  4506  4546  4583  4613  4735
.TITLE      1645


 ERRORS DETECTED:  0
 DEFAULT GLOBALS GENERATED:  0

 *,DERSAA.SEQ/SOL/CRF/PAGNUM/NL:TOC=DERSAA.SML,DERSAA.P11
 RUN-TIME:  29 46 7 SECONDS
 RUN-TIME RATIO: 196/83=2.3
 CORE USED:  22K  (43 PAGES)
```