RP04

DISKLESS CONTR TEST 1

MD-11-DERPS-B

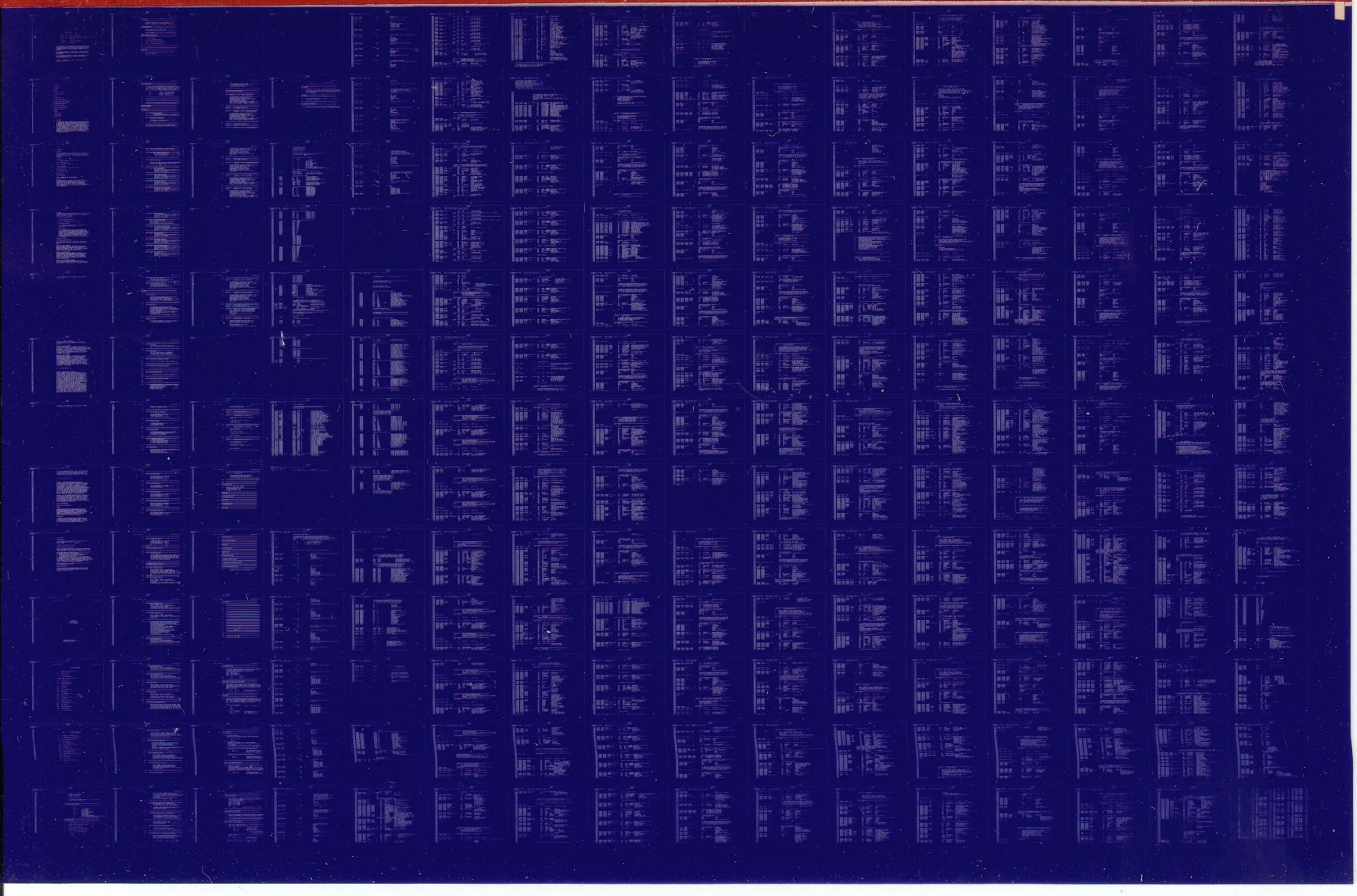
EP-DERPS-B-DL-A

FICHE 1 OF 2

NOV 1976

digital

MADE IN US



RP04

STATIC TEST NO. 1

MD-11-DERPS-B

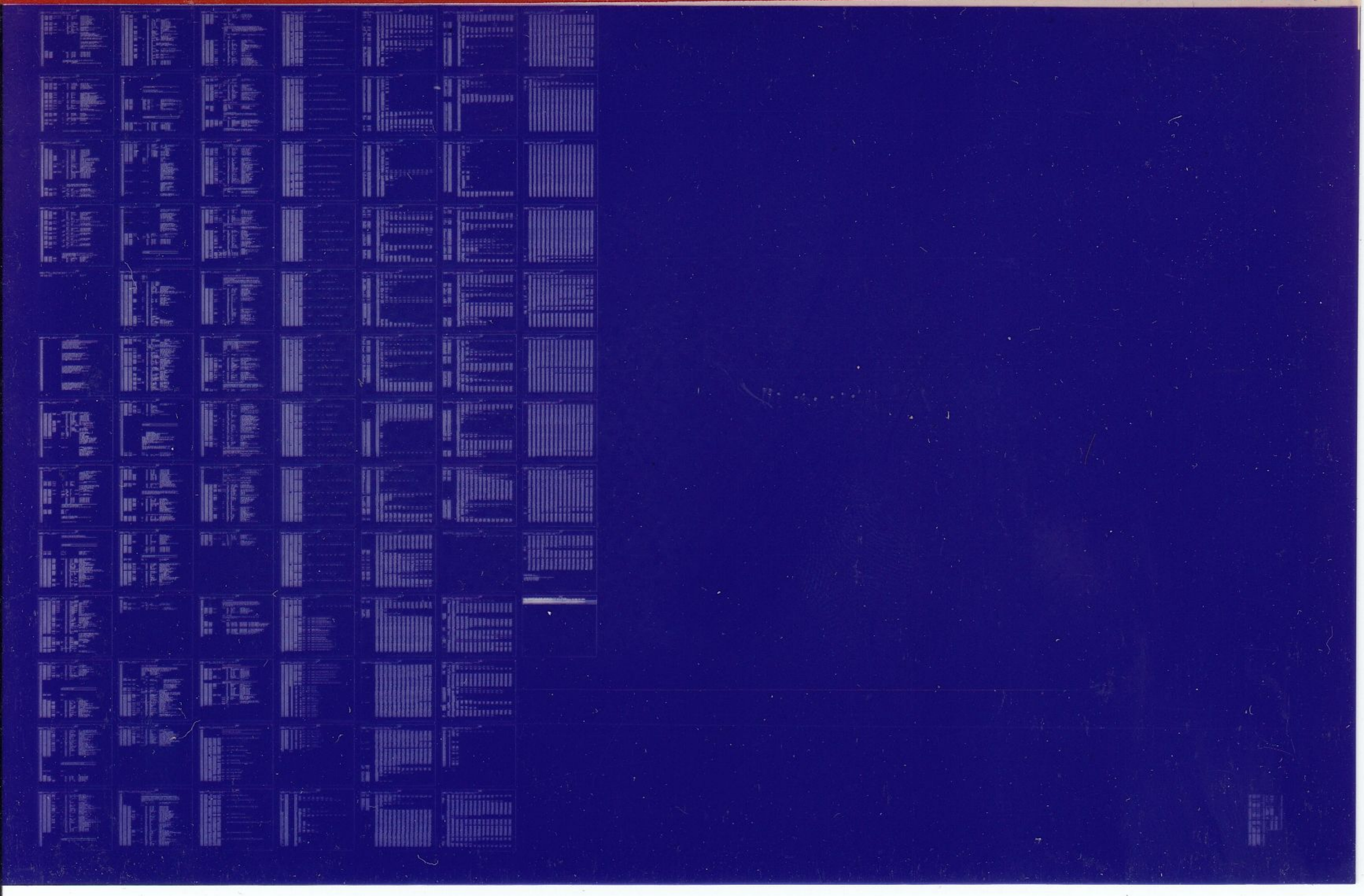
EP-DERPS-B-DL-A

FICHE 2 OF 2

NOV 1976

digital

MADE IN USA



.REM %

IDENTIFICATION

PRODUCT CODE: MAINDEC-11-DERPS-B-D
PRODUCT NAME: RP04 DISKLESS CONTROLLER TEST-PART I (STATIC 1A)
DATE: APRIL 1976
MAINTAINER: DIAGNOSTIC ENGINEERING
AUTHOR: SUB MALLICK

CONTENTS

1. ABSTRACT
2. REQUIREMENTS
2.1 EQUIPMENT
2.2 STORAGE
2.3 PRELIMINARY PROGRAMS
3. LOADING PROCEDURE
3.1 METHOD
4. STARTING PROCEDURE
4.1 CONTROL SWITCH SETTINGS
4.2 STARTING ADDRESS OR ADDRESSES
4.3 PROGRAM AND/OR OPERATOR ACTION
5. OPERATING PROCEDURE
5.1 OPERATIONAL SWITCH SETTINGS
5.2 SUB-ROUTINE ABSTRACTS
6. ERRORS
7. RESTRICTIONS
8. MISCELLANEOUS
8.1 EXECUTION TIME

1.0 ABSTRACT

THIS DIAGNOSTIC TESTS THE RH70 AND DCL OF AN RP04 SUBSYSTEM.
IT DOES NOT USE THE DISK SURFACE OR ANY SIGNALS FROM THE MDLI.
IT REQUIRES THAT THE DCL CABLE BE PLUGGED INTO THE MDLI OR
BE APPROPIATELY TERMINATED. IF THE DISK IS POWERED UP, IT
IS REQUIRED TO GET THE DISK TO THE "HEADS UNLOADED" POSITION.
AFTER A SUCCESSFUL RUN (WITH NO ERRORS) OF THIS DIAGNOSTIC
IT CAN BE ASSERTED THAT, "THAT PART OF THE DCL THAT HANDLES
DATA OR DATA ASSOCIATED LOGIC IS WORKING PROPERLY". THIS
IMPLIES THAT, THAT PART OF THE LOGIC WHICH HANDLES MECHANICAL
COMMANDS OR ITS ASSOCIATED LOGIC IS NOT TESTED IN THIS
DIAGNOSTIC. ALL DATA COMMANDS USE THE MAINTENANCE REGISTER
IN THE WRAPAROUND MODE.

```
                                2.0     REQUIREMENTS

                                2.1     EQUIPMENT

                                        PDP-11 COMPUTER WITH CONSOLE TELETYPE, AND A RP04 DISK SYSTEM.
                                        THE RP04 DISK SYSTEM WILL CONSIST OF AN RH70 CONTROLLER,
                                        A DISK CONTROL LOGIC (DCL), THE CABLE FROM THE DCL CAN BE
                                        CONNECTED TO THE MDLI BUT IF NOT THAT CABLE MUST BE PROPERLY
                                        TERMINATED.

                                2.2     STORAGE

                                        THIS PROGRAM REQUIRES 16K WORDS OF MEMORY.

                                2.3     PRELIMINARY PROGRAMS

                                        THIS IS THE FIRST PROGRAM TO RUN

                                3.0     LOADING PROCEDURE

                                        USE STANDARD PROCEDURE FOR LOADING .ABS TAPES

                                4.0     STARTING PROCEDURE

                                4.1     CONTROL SWITCH SETTINGS

                                        SEE SECTION 5.1

                                4.2     STARTING ADDRESS

                                        START AT ADDRESS 200---FOR NORMAL RUN
                                        START AT ADDRESS 210---FOR UNIT SELECTION

                                        200 START
                                        ALL SWITCHES MUST BE DOWN FOR WORST CASE RUN.  WITH THIS
                                        STARTING ADDRESS ALL THE RP04S ON THE SYSTEM WILL BE
                                        TESTED ONE AT A TIME BEFORE "END PASS" IS PRINTED OUT.
                                        TESTING WILL START WITH THE LOWEST UNIT NUMBER DRIVE
                                        THAT IS POWERED UP (THAT IS THE LOWEST UNIT NUMBER RHAS REGISTER
                                        THAT RESPONDS)THEN GO ON TO THE NEXT HIGHER UNIT NUMBER
                                        THAT IS POWERED UP.
```

```
                                        210 START
                                        ALL SWITCHES MUST BE DOWN FOR WORST CASE RUN.  WITH THIS
                                        STARTING ADDRESS THE CONSOLE TELETYPE WILL ASK FOR THE UNIT
                                        NUMBER TO BE TESTED.  THEN ONLY THAT UNIT WILL BE TESTED
                                        FOR EACH PASS OF THE PROGRAM.

                                4.3     PROGRAM AND/OR OPERATOR ACTION

                                        1.  LOAD THE PROGRAM INTO MEMORY.

                                        2.  SET STARTING ADDRESS ON THE SWITCH REGISTER

                                        3.  PRESS "LOAD ADDRESS".

                                        4.  SET "OPERATIONAL SWITCH SETTINGS" (SEE SECTION 5.1)
                                            WORST CASE IS ALL SWITCHES DOWN.

                                        5.  PRESS "START".

                                        6.  FOR THE FIRST PASS EACH TEST WILL BE EXECUTED ONCE
                                            ON THE DRIVES PRESENT OR DRIVE SELECTED BEFORE "END
                                            PASS" IS PRINTED.  THE FIRST PASS WILL REQUIRE OPERATOR
                                            INTERVENTION IF THE PROGRAM IS NOT RUN UNDER AN "ACT-11"
                                            MONITOR.  THE SECOND AND SUBSEQUENT PASSES WILL EXECUTE
                                            EACH TEST FOUR TIMES ON EACH DRIVES PRESENT OR DRIVE
                                            SELECTED BEFORE "END PASS" IS PRINTED.  THE SECOND
                                            AND SUBSEQUENT PASSED DO NOT NEED ANY OPERATOR INTERVENTION.

                                5.0     OPERATING PROCEDURE

                                5.1     OPERATIONAL SWITCH SETTINGS

                                        SWITCH DEFINITIONS ARE GIVEN IN SECTION 9 "OPERATIONAL
                                        SWITCH SETTINGS" HOWEVER THE DETAIL DESCRIPTION ARE GIVEN
                                        HERE.

                                        SWITCH 15 - HALT ON ERROR
                                        WHEN THIS SWITCH IS SET, IF THE PROGRAM FINDS AN ERROR
                                        THEN THE APPROPIATE INFORMATION WILL BE PRINTED OUT
                                        AND THEN THE PROGRAM WILL HALT.  AFTER THIS HALT, PRESSING
                                        "CONTINUE" WILL CONTINUE WITH THE PROGRAM TILL THE NEXT
                                        ERROR IS FOUND WHEN THE SAME THING WILL HAPPEN.

                                        SWITCH 14 - LOOP ON TEST
                                        WHEN THIS SWITCH IS SET THE PROGRAM WILL BEGIN TO LOOP
                                        ON THE CURRENT TEST BEING EXECUTED.  FOR EXAMPLE IF THIS
                                        SWITCH IS SET WHEN THE PROGRAM IS IN TEST 10 THEN THE
                                        PROGRAM WILL KEEP EXECUTING ALL OF TEST 10 REPEATEDLY.
                                        ONE WAY TO BE SURE THAT THE PROGRAM IS IN THE EXPECTED
                                        TEST IS TO SET THIS SWITCH DURING AN ERROR PRINTOUT OR
                                        DURING A PROGRAM HALT.

                                        SWITCH 13 - INHIBIT ERROR TYPEOUTS
                                        WHEN THIS SWITCH IS SET FURTHER ERROR PRINTOUTS WILL
                                        CEASE, HOWEVER OPERATOR INSTRUCTIONS SUCH AS "STOP DRIVE X"
                                        WILL CONTINUE.  AT THE END OF PASS "TOTAL NUMBER OF ERRORS
```

```
                                ON THIS PASS ON DRIVE X" WILL BE TRUE, THAT IS, ALTHOUGH
                                PRINTOUTS WERE INHIBITED IF THAT PASS FOUND 6 ERRORS,
                                IT WILL SAY SO.
```

```
                        SWITCH 11 - INHIBIT ITERATIONS
                        WHEN THIS SWITCH IS SET THE PROGRAM ON SECOND PASS WILL
                        NOT REPEAT EACH TEST FOUR TIMES BUT WILL DO EACH TEST
                        ONCE ONLY.

                        SWITCH 10 - BELL ON ERROR
                        WHEN THIS SWITCH IS SET, IF THE PROGRAM FINDS AN ERROR
                        THE "BELL" OR "ALARM" WILL BE SOUNDED.  THIS SWITCH IS USEFUL
                        WHEN SWITCH 11 IS SET YET INFORMATION IS NEEDED WHEN ANY ERROR
                        IS DETECTED. TAKE THE EXAMPLE OF A PROGRAM LOOPING ON A TEST WITH
                        SWITCH 11 SET TO HELP SCOPING.  THEN IF THIS SWITCH IS
                        SET AND THE BELL OR ALARM SOUNDS IT MEANS THAT THE ERROR
                        IS PRESENT BUT IF THE BELL OR ALARM STOPS IT MEANS THAT
                        THE ERROR IS NOT PRESENT.

                        SWITCH 9 - LOOP ON ERROR
                        WHEN THIS SWITCH IS SET, IF THE PROGRAM FINDS AN ERROR
                        THEN GENERALLY THE PROGRAM WILL LOOP BACK TO THE LAST
                        EXECUTED "SCOPE" STATEMENT.  IF ON THE SECOND TIME
                        THROUGH AN ERROR IS FOUND IT WILL AGAIN LOOP BACK TO
                        THAT "SCOPE" STATEMENT.  THIS LOOPING WILL CONTINUE AS LONG
                        AS THE ERROR IS PRESENT AND THIS SWITCH IS SET.  HOWEVER
                        IF THE ERROR IS NOT PRESENT AT ANY TIME THEN IT WILL
                        CONTINUE NORMALLY WITH THE PROGRAM.  EACH TIME THE ERROR
                        IS ENCOUNTERED PRINTOUT WILL TAKE PLACE UNLESS SWITCH 11
                        IS ALSO SET.  DURING BEGUG, USING A SCOPE, IT IS RECOMMENDED
                        THAT SWITCH 11 IS ALSO SET.

                        NOTE: ALSO SEE SECTION 8.3


                        SWITCH 8 - LOOP ON TEST IN SWR <7:0>
                        THIS IS A SPECIAL SWITCH.  WHEN SET SWITCHES 0 THRU 7
                        HAVE ONE MEANING AND WHEN RESET SWITCHES 0 THRU 7 HAVE
                        ANOTHER MEANING.  THIS MEANS THAT ANY SETTING OF SWITCH
                        0 THRU 7 MUST BE DONE WITH SWITCH 8 IN THE APPROPIATE
                        POSITION.  WHEN THIS SWITCH IS SET THEN SWITCHES 0 THRU
                        7 GIVE THE TEST NUMBER TO BE LOOPED ON.  FOR EXAMPLE
                        WITH SWITCH 8 SET AND SWITCH 3 SET THE PROGRAM WILL LOOP
                        ON TEST 10.  HOWEVER THIS SETTING MUST BE DONE AT THE
                        BEGINNING OF THE PROGRAM THEN ALL THE TESTS FROM 1 TO 10
                        WILL BE EXECUTED AND THEN TEST 10 WILL BE REPEATED OVER
                        AND OVER AGAIN.  WHEN THIS SWITCH IS NOT SET THEN SWITCHES
                        0 THRU 7 HAVE THE MEAING ITS NAME INDICATES.
                        FOR EXAMPLE SWITCH 7 IS "STOP FURTHER COMPARES: THAT IS
                        IF SWITCH 8 IS NOT SET AND SWITCH 7 IS SET THEN WHEN A
                        DATA ERROR IS DETECTED NO FURTHER COMPARES WILL BE DONE.
                        FOR EXAMPLE IN A 256 WORD BUFFER IF ALL THE WORDS ARE IN
                        ERROR THEN AFTER SEEING THE PRINTOUT FOR THE FIRST FEW
                        WORDS SETTING SWITCH 7 ONLY WILL STOP FURTHER PRINTOUTS
                        OF THIS ERROR AND GO ON WITH THE TEST RATHER THAN PRINT
                        ALL THE 256 WORDS.  HOWEVER IF THIS WAS DONE WITH SWITCH 11
                        THEN THE NEXT ERROR THAT THE PROGRAM DETECTS IN A SUB-
                        SEQUENT TEST WILL ALSO BE LOST.  BUT WITH SWITCH 7, ONLY
```

```
                                THIS GROUP OF DATA ERRORS ARE NOT PRINTED OUT.  ANOTHER
                                EXAMPLE OF SWITCH 8 BEING LOW IS WITH SWITCH 6, WHICH
```

IS "ECC TEST-COMPARE END RESULT ONLY". THAT IS IF SWITCH
8 IS NOT SET AND SWITCH 6 IS SET THEN ON ECC TESTS (TEST
120 THRU TEST 134) INSTEAD OF COMPARING CONTENTS OF THE
POSITION REGISTER AND PATTERN REGISTER AFTER EVERY CLOCK,
COMPARES WILL ONLY BE DONE AT THE END OF ALL THE CLOCKS.

NOTE: ALSO SEE SECTION 8.3

SWITCH 7 - STOP FURTHER COMPARES IF SW08 IS LOW.
IF SWITCH 8 IS SET AND THIS SWITCH IS ALSO SET THEN THIS
SWITCH GIVES THE TEST NUMBER TO BE LOOPED ON AS INDICATED
IN THE DESCRIPTION OF SWITCH 8. IF SWITCH 8 IS NOT SET
AND THIS SWITCH IS SET THEN THE PROGRAM WILL DO AS THE
NAME INDICATES. FOR EXAMPLE IN A 256 WORD BUFFER IF ALL
THE WORDS ARE IN ERROR THEN AFTER SEEING THE ERROR PRINTOUTS
FOR THE FIRST FEW WORDS THEN SETTING SWITCH 7 WITH SWITCH 8
NOT SET WILL STOP THE PRINTOUT OF ALL 256 WORDS BUT WILL
NOT STOP THE PRINTOUT OF ANOTHER ERROR IN ANY SUBSEQUENT
TEST. IT IS EXPECTED THAT SWITCH 7 AFTER BEING SET FOR
A WHILE TO STOP PRINTING ALL THE 256 WORDS WILL BE RESET
AGAIN TO ENABLE THE PRINTING OF OTHER DATA ERRORS.

SWITCH 6 - ECC TEST-COMPARE END RESULTS ONLY IF SW08 IS LOW
IF SWITCH 8 IS SET AND THIS SWITCH IS ALSO SET THEN THIS
SWITCH GIVES THE TEST NUMBER TO BE LOOPED ON AS INDICATED
IN THE DESCRIPTION OF SWITCH 8. IF SWITCH 8 IS NOT SET
AND THIS SWITCH IS SET THEN ON ECC TESTS (TEST 120 THRU
TEST 134) INSTEAD OF COMPARING CONTENTS OF THE POSITION AND
PATTERN REGISTERS AFTER EVERY CLOCK, COMPARES WILL BE DONE
ONLY AT THE END OF ALL THE CLOCKS.

5.2 SUB-ROUTINE ABSTRACTS

SEE SECTION 9 "SUBROUTINES"

6.0 ERRORS

ERROR PRINTOUTS CONTAIN THE ERROR ADDRESS AND OTHER
PERTINENT INFORMATION CONCERNING THE PARTICULAR FAILURE.
THIS INFORMATION MAY BE THE CONTENTS OF RELEVANT RP04
REGISTERS OR GOOD/RECEIVED DATA. IF THE ERROR OCCURRED IN A
SUBROUTINE, THE ADDRESS OF THE SUBROUTINE CALL IS ALSO
GIVEN. REFER TO THE PROGRAM LISTING AT THE STATED
ADDRESS TO DETERMINE THE CAUSE OF THE ERROR.

7.0 RESTRICTIONS

IF THERE IS A DRIVE CONNECTED THEN THE OPERATOR MUST HAVE
THE DRIVE PORT SWITCH LOCKED EITHER ON PORT A OR PORT B
BUT NEVER LEAVE IT IN THE PROGRAMMABLE STATE. IF THERE
IS NO DRIVE CONNECTED THEN THE CABLE NORMALLY GOING FROM
THE DCL TO THE MDLI MUST BE PROPERLY TERMINATED.

8.0 MISCELLANEOUS

8.1 EXECUTION TIME

THE FIRST PASS OF THE PROGRAM WILL TAKE 1.75 MINUTES PER DRIVE. SUBSEQUENT PASSES WILL TAKE 7 MINUTE.

8.2 STACK POINTER

THE STACK IS INITIALLY SET TO 1000

8.3 OPERATOR SELECTABLE SCOPE LOOPS

HERE IS A DETAILED EXPLAINATION OF HOW THE LOOP ON ERROR WORKS.
ON HITTING AN ERROR IF THE LOOP ON ERROR SWITCH IS SET, THE PROGRAM GOES BACK - USUALLY BACK TO THE BEGINNING OF THE TEST.

WHEN THIS OPERATOR SELECTABLE SCOPE LOOP IS USED THEN THE POINT THE PROGRAM GOES BACK TO CAN BE CHANGED.
THE RESTRICTIONS TO THE POINT WHERE THE PROGRAM CAN GO ARE: -
1. IT MUST BE WITHIN THE TEST UNDER CONSIDERATION
2. LOOP ON ERROR SWITCH MUST BE SET
3. THE ERROR MUST OCCUR WITHIN THE TEST UNDER CONSIDERATION
IF THE ERROR DOES NOT OCCUR WITHIN THE TEST UNDER CONSIDERATION THE PROGRAM WILL REVERT TO NORMAL OPERATION. HOWEVER, IF LOOP ON TEST SWITCH IS SET AND THIS OPERATOR SELECTABLE SCOPE LOOP IS USED THEN THE PROGRAM WILL LOOP BACK TO THE SELECTED POINT WHEN IT COMES TO THE END OF THE TEST UNDER CONSIDERATION.

AFTER LOOPING FOR SOME TIME IF THE LOOP SWITCH IS PUT DOWN THEN NORMAL OPERATION WILL CONTINUE.

9.0 PROGRAM DESCRIPTION

THE FOLLOWING SECTIONS DESCRIBE EACH TEST AND SUBROUTINES IN DETAIL AND CAN ALSO BE USED AS AN INDEX TO THE LISTING. THE LEFT MOST COLUMN IS THE LINE NUMBER WITHIN THE LISTING WHERE THAT ITEM WILL BE FOUND.

DOCUMENT
****
MAINDEC-11-DERPS-B
****

TABLE OF CONTENTS
****

13	OPERATIONAL SWITCH SETTINGS
27	BASIC DEFINITIONS
133	TRAP CATCHER
140	STARTING ADDRESS(ES)
151	MEMORY MANAGEMENT DEFINITIONS
190	COMMON TAGS
246	ERROR POINTER TABLE
897	REGISTER ADDRESSES
1063	REGISTER TEST
2156	SILO TEST
7882	END OF PASS ROUTINE
7924	SUBROUTINES
7990	SAVE REGISTERS ROUTINE
8018	FLOAT 1 AND 0
8073	CLEAR MEMORY ROUTINE
8106	LOCAL TRAPS
8123	CLEAD DISK ROUTINE
8136	CHECK DISK STATUS ROUTINE
8263	SAVE ROUTINE
8288	WRITE CHECK ROUTINE
8324	COMPARE ROUTINE
8414	CRC GENERATION ROUTINE

TABLE OF CONTENTS
****

8727	JAM CURRENT CYLINDER ROUTINE
8764	ECC GENERARION AND COMPARISON ROUTINE
9100	RH BASE ADDRESS CHANGE ROUTINE
9172	DISK SIMULATION
10141	SCOPE HANDLER ROUTINE
10215	CONVERT BINARY TO DECIMAL AND TYPE ROUTINE
10283	TYPE ROUTINE
10359	TTY INPUT ROUTINE
10470	READ AN OCTAL NUMBER FROM THE TTY
10524	ERROR HANDLER ROUTINE
10570	ERROR MESSAGE TYPEOUT ROUTINE
10628	BINARY TO OCTAL (ASCII) AND TYPE
10706	TRAP DECODER
10721	TRAP TABLE
10745	POWER DOWN AND UP ROUTINES

```
                                MAINDEC-11-DERPS-B                                  DECDOC VER  00.04

                                        2       COPYRIGHT (C) 1975,1976
                                                DIGITAL EQUIPMENT CORP.
                                                MAYNARD, MASS. 01754

                                                PROGRAM BY SUB MALLICK

                                                THIS PROGRAM WAS ASSEMBLED USING THE PDP-11 MAINDEC SYSMAC
                                                PACKAGE (MAINDEC-11-DZQAC-A5).


                                        ;*******************************************************************
                                    13  ;OPERATIONAL SWITCH SETTINGS
                                        ;*******************************************************************

                                        14
                                                        ;SWITCH                 USE
                                                        ;------         -------------------
                                                          15            HALT ON ERROR
                                                          14            LOOP ON TEST
                                                          13            INHIBIT ERROR TYPEOUTS
                                                          11            INHIBIT ITERATIONS
                                                          10            BELL ON ERROR
                                                           9            LOOP ON ERROR
                                                           8            LOOP ON TEST IN SWR<7:0>
                                                           7            STOP FURTHER COMPARES IF SW08 IS LOW
                                                           6            ECC TEST-COMPARE END RESULTS ONLY IF SW0


                                        ;*******************************************************************
                                    27  ;BASIC DEFINITIONS
                                        ;*******************************************************************

                                        29      INITIAL ADDRESS OF THE STACK POINTER *** 1000 ***

                                        40      GENERAL PURPOSE REGISTER DEFINITIONS

                                        52      PRIORITY LEVEL DEFINITIONS

                                        62      "SWITCH REGISTER" SWITCH DEFINITIONS

                                        90      DATA BIT DEFINITIONS (BIT00 TO BIT15)

                                        118     BASIC "CPU" TRAP VECTOR ADDRESSES
```

```
                        ;*****************************************************************
                        ;TRAP CATCHER
                        ;*****************************************************************
                        ;
                        ;ALL UNUSED LOCATIONS FROM 4 - 776 CONTAIN A ".+2,HALT"
                        ;SEQUENCE TO CATCH ILLEGAL TRAPS AND INTERRUPTS
                        ;LOCATION 0 CONTAINS 0 TO CATCH IMPROPERLY LOADED VECTORS
                        ;
                        ;*****************************************************************
                        ;STARTING ADDRESS(ES)
                        ;*****************************************************************
                        ;
                        ;STARTING ADDRESS 200 FOR NORMAL STARTS
                        ;THIS WILL TEST ALL RP04'S ON THE SYSTEM A SINGLE DRIVE AT A TIME
                        ;
                        ;STARTING ADDRESS 210 WILL TEST ONLY ONE SPECIFIED DRIVE
                        ;
                        ;*****************************************************************
                        ;MEMORY MANAGEMENT DEFINITIONS
                        ;*****************************************************************
                        ;
                        ;KT11 VECTOR ADDRESS
                        ;
                        ;KT11 STATUS REGISTER ADDRESSES
                        ;
                        ;KERNAL "I" PAGE DESCRIPTOR REGISTERS
                        ;
                        ;KERNAL "I" PAGE ADDRESS REGISTERS
                        ;
                        ;*****************************************************************
                        ;
                        ;*****************************************************************
                        ;COMMON TAGS
                        ;*****************************************************************
                        ;
                        ;THIS TABLE CONTAINS VARIOUS COMMON STORAGE LOCATIONS
                        ;USED IN THE PROGRAM.
                        ;
                        ;*****************************************************************
```

```
                                MAINDEC-11-DERPS-B                                      DECDOC VER  00.04

                        ;*******************************************************************
                246     ;ERROR POINTER TABLE
                        ;*******************************************************************

                248     ;       THIS TABLE CONTAINS THE INFORMATION FOR EACH ERROR THAT CAN OCCU
                        ;       THE INFORMATION IS OBTAINED BY USING THE INDEX NUMBER FOUND IN
                        ;       LOCATION $ITEMB. THIS NUMBER INDICATES WHICH ITEM IN THE TABLE I
                        ;       NOTE1:  IF $ITEMB IS 0 THE ONLY PERTINENT DATA IS ($ERRPC).
                        ;       NOTE2:  EACH ITEM IN THE TABLE CONTAINS 4 POINTERS EXPLAINED AS

                254     ;               EM              ;;POINTS TO THE ERROR MESSAGE
                        ;               DH              ;;POINTS TO THE DATA HEADER
                        ;               DT              ;;POINTS TO THE DATA
                        ;               DF              ;;POINTS TO THE DATA FORMAT

                263     ;*******************************************************************

                700     ;*******************************************************************

                705     ;*******************************************************************

                739     ;*******************************************************************
                        ;*******************************************************************

                        ;*******************************************************************
                897     ;REGISTER ADDRESSES
                        ;*******************************************************************

                        ;*******************************************************************
               1063     ;REGISTER TEST
                        ;*******************************************************************

               1147     ;*******************************************************************
                        ;TEST 1  REFERENCE EACH REGISTER
                        ;                       REFERENCE EACH REGISTER BY A MOVE INSTRUCTION
                        ;*******************************************************************

               1191     ;*******************************************************************
                        ;TEST 2  RHCS2-CONTROL AND STATUS 2

               1194     ;               THIS PARTIALLY TESTS RHCS2 TO ENABLE DETERMINATION
                        ;               OF THE NUMBER OF DRIVES PRESENT

               1197     ;*******************************************************************

               1213     ;*******************************************************************
                        ;TEST 3  PARTIAL TEST FOR RHAS FOR UNIT NUMBERS PRESENT
                        ;*******************************************************************
```

```
                1233    ;*****************************************************************
                        ;TEST 4  TEST FOR DRIVES PRESENT USING RHAS AND RHCS2
                        ;*****************************************************************

                1343    ;*****************************************************************
                        ;TEST 5  TEST SERIAL NUMBER AND DRIVE TYPEI
                        ;               READ SERIAL NUMBER REGISTER AND DRIVE TYPE REGIS
                        ;               TYPE IT OUT AND PROCEED
                        ;               TO LOOP HERE SET SWITCH 8 AND THIS TEST NO AND R

                1349    ;*****************************************************************

                1385    ;*****************************************************************
                        ;TEST 6  CHECK MOL TO BE LOW

                1388    ;       MAKE SURE THAT DRIVE IS OFF LINE BEFORE STARTING PROGRAM
                        ;       IF DRIVE IS ON LINE THEN AFTER TYPE OUT THE PROGRAM WILL
                        ;       HANG FOR EVER WAITING FOR DRIVE TO GO OFF LINE

                1392    ;*****************************************************************

                1423    ;*****************************************************************
                        ;TEST 7  RHWC - WORD COUNT REGISTER
                        ;               TEST LOADING AND READING OF ALL POSSIBLE BITS IN
                        ;               REGISTERS.  USE A PATTERN OF WALKING 0'S (-2,-3,
                        ;               WALKING 1'S (1,2,4,10 ETC)
                        ;*****************************************************************

                1449    ;*****************************************************************
                        ;TEST 10 RHBA - UNIBUS ADDRESS REGISTER
                        ;               TEST LOADING AND READING OF ALL POSSIBLE BITS IN
                        ;               REGISTERS.  USE A PATTERN OF WALKING 0'S (-2,-3,
                        ;               WALKING 1'S (1,2,4,10 ETC)
                        ;*****************************************************************

                1475    ;*****************************************************************
                        ;TEST 11 RHCS2 - CONTROL AND STATUS 2
                        ;               TEST LOADING AND READING OF ALL POSSIBLE BITS IN
                        ;               REGISTERS.  USE A PATTERN OF WALKING 0'S (-2,-3,
                        ;               WALKING 1'S (1,2,4,10 ETC)
                        ;*****************************************************************

                1502    ;*****************************************************************
                        ;TEST 12 RHCS1 - CONTROL AND STATUS 1 REGISTER
                        ;               TEST LOADING AND READING OF ALL POSSIBLE BITS IN
                        ;               REGISTERS.  USE A PATTERN OF WALKING 0'S (-2,-3,
                        ;               WALKING 1'S (1,2,4,10 ETC)
                        ;*****************************************************************
```

```
                                MAINDEC-11-DERPS-B                                      DECDOC VER  00.04

                        1528    ;*****************************************************************
                                ;TEST 13 RHER1 - ERROR REGISTER #1
                                ;               TEST LOADING AND READING OF ALL POSSIBLE BITS IN
                                ;               REGISTERS.  USE A PATTERN OF WALKING 0'S (-2,-3,
                                ;               WALKING 1'S (1,2,4,10 ETC)
                                ;*****************************************************************

                        1557    ;*****************************************************************
                                ;TEST 14 RHMR - MAINTENANCE REGISTER
                                ;       BIT 0 (DMD) MUST BE SET BEFORE THE OTHER BITS
                                ;       ARE READ WRITE
                                ;       ONLY 5 LOW ORDER BITS ARE TESTED (R2 HAS 5)
                                ;*****************************************************************

                        1616    ;*****************************************************************
                                ;TEST 15 RHDST - DESIRED SECTOR/TRACK ADDRESS
                                ;               TEST LOADING AND READING OF ALL POSSIBLE BITS IN
                                ;               REGISTERS.  USE A PATTERN OF WALKING 0'S (-2,-3,
                                ;               WALKING 1'S (1,2,4,10 ETC)
                                ;*****************************************************************

                        1642    ;*****************************************************************
                                ;TEST 16 RHER2 - ERROR REGISTER #2
                                ;               TEST LOADING AND READING OF ALL POSSIBLE BITS IN
                                ;               REGISTERS.  USE A PATTERN OF WALKING 0'S (-2,-3,
                                ;               WALKING 1'S (1,2,4,10 ETC)
                                ;*****************************************************************

                        1668    ;*****************************************************************
                                ;TEST 17 RHOF - MARGIN/OFFSET REGISTER
                                ;               TEST LOADING AND READING OF ALL POSSIBLE BITS IN
                                ;               REGISTERS.  USE A PATTERN OF WALKING 0'S (-2,-3,
                                ;               WALKING 1'S (1,2,4,10 ETC)
                                ;*****************************************************************

                        1694    ;*****************************************************************
                                ;TEST 20 RHCA - DESIRED CYLINDER REGISTER
                                ;               TEST LOADING AND READING OF ALL POSSIBLE BITS IN
                                ;               REGISTERS.  USE A PATTERN OF WALKING 0'S (-2,-3,
                                ;               WALKING 1'S (1,2,4,10 ETC)
                                ;*****************************************************************

                        1720    ;*****************************************************************
                                ;TEST 21 RHER3 - ERROR REGISTER #3
                                ;               TEST LOADING AND READING OF ALL POSSIBLE BITS IN
                                ;               REGISTERS.  USE A PATTERN OF WALKING 0'S (-2,-3,
                                ;               WALKING 1'S (1,2,4,10 ETC)
                                ;*****************************************************************
```

```
                                MAINDEC-11-DERPS-B                                                DECDOC VER  00.04

                        1756    ************************************************************************
                                TEST 22 RHCS1 - BIT # 13 - MCPE
                                        THIS FORCES A MASS BUS CONTROL PARITY ERROR
                                        BY SETTING PAT AND READING RHER1
                                ************************************************************************

                        1812    ************************************************************************
                                TEST 23 CONTROL AND STATUS 2 (RHCS 2)
                                        THIS TESTS THE UNIT SELECT BIT #0-2 (US1-4) CLEAR BIT #5
                                        NON-EXISTENT DRIVE BIT#12 (NED)
                                        THE OTHER RHCS2 BITS ARE NOT TESTED HERE

                        1818    ************************************************************************

                                ************************************************************************
                        2156    SILO TEST
                                ************************************************************************

                        2159    ************************************************************************
                                TEST 24 SILO TST 1

                        2162            THIS TESTS THE SILO BUFFER IN THE RH11 CONTROLLER
                                        A READ IS ATTEMPTED FROM AN EMPTY SILO
                                        DATA LATE (DLT) (RHCS2), TRANSFER ERROR (TRE) (RHCS1),
                                        SPECIAL CONDIIION (SC) (RHCS1) SHOULD SET
                                        THEN LOADING "1" INTO TRE SHOULD CLEAR DLT, TRE AND SC

                        2168    ************************************************************************

                        2216    ************************************************************************
                                TEST 25 SILO TEST 2

                        2219            THIS TESTS THE IR AND "OR" BITS OF RHCS2
                                        AT THE BEGINNING IR SHOULD BE SET AND "OR" RESET
                                        LOADING 0 IN SILO RESETS IR FOR ONLY 2 MICRO SECONDS
                                        THIS TIME CANNOT BE CHECKED BUT IT IS CHECKED TO SEE IF
                                        IT DOES GO DOWN OR NOT
                                        THEN ALL 1 IS LOADED IN SILO "OR" SHOULD BECOME SET
                                        IN 30 MICRO SECONDS AGAIN TIME IS NOT  CHECKED
                                        "OR" SHOULD BE SET
                                        THE OUTPUT FROM THE SILO SHOULD BE 0 AND ALL ONES

                        2229    ************************************************************************

                        2282    ************************************************************************
                                TEST 26 SILO TEST 3

                        2285            THIS TESTS SILO BUFFER BY FILLING IT WITH A COUNT FROM
                                        0 TO 65 AND THEN CHECKING IF IR IS DOWN AND "OR"
                                        IS HIGH AND COMPARING THE SILO OUTPUT.
```

```
                                MAINDEC-11-DERPS-B                                                DECDOC VER  00.04

                                2289    ;*****************************************************************
                                2351    ;*****************************************************************
                                        TEST 27 SILO TEST4
                                                NOW PUT 67 WORDS INTO SILO AND CHECK FOR DLT
                                                EVEN AFTER THE 67TH. WORD INPUT THE FIRST WORD SHOULD NO
                                2356    ;*****************************************************************
                                2386    ;*****************************************************************
                                        TEST 30 SILO TEST 5
                                                THE SILO IS LOADED WITH 0,1,2,3 THEN AFTER
                                                'OR' IS UP A CLR IN RHCS2 IS DONE THEN 4,
                                                IS LOADED. AFTER 'OR' IS UP 2 READS FROM'
                                                SILO IS DONE ON THE LAST DTL IN RHCS2 SHOULD BE SET
                                2393    ;*****************************************************************
                                2468    ;*****************************************************************
                                        TEST 31 TEST ODD BYTE INSTRUCTION ON RHCS1
                                2471            RDY (BIT 07) AND DVA (BIT 11) SHOULD ALWAYS BE SET
                                2473    ;*****************************************************************
                                2509    ;*****************************************************************
                                        TEST 32 TEST ODD BYTE INSTRUCTION ON RHCS2
                                2512            IR (BIT 06) AND THE UNIT SELECT (BIT 0-2) WILL BE SET
                                2514    ;*****************************************************************
                                2556    ;*****************************************************************
                                        TEST 33 ODD BYTE TEST ON RHWC
                                2559            IN THIS REGISTER NO BITS SHOULD BE PERMANENTLY SET
                                2562    ;*****************************************************************
                                2593    ;*****************************************************************
                                        TEST 34 TEST ODD BYTE INSTRUCTION ON RHBA
                                2596            BIT 0 SHOULD ALWAYS BE 0
                                2598    ;*****************************************************************
                                2630    ;*****************************************************************
                                                FOUR GENERAL REGISTERS WILL BE RESERVED FOR HARDWARE
                                                R1=RHCS1 CONTROL AND STATUS1
                                                R2=RHCS2 CONTROL AND STATUS2
                                                R3=RHDS1 DRIVE STATUS 1
                                                R4=RHER1 ERROR REGISTER1
```

```
                                MAINDEC-11-DERPS-B                                  DECDOC VER  00.04

                        2637            WHENEVER ANY OTHER USE IS MADE OF THESE REGISTERS
                                        APPROPIATE SAVING MUST BE DONE

                        2640    ;*******************************************************************

                        2648    ;*******************************************************************
                                TEST 35 PACK ACKNOWLEDGE COMMAND TEST

                        2651            THE PACK ACKNOWLEDGE COMMAND WILL BE LOADED INTO RHCS1 W
                                        THEN ALL REGISTERS WILL BE CHECKED
                                        RH CLEAR WILL BE GIVEN
                                        THEN ALL REGISTERS WILL BE CHECKED

                        2656    ;*******************************************************************

                        2721    ;*******************************************************************
                                TEST 36 UNIBUS INIT TEST

                        2724            ALL POSSIBLE REGISTERS ARE FILLED WITH ONES
                                        A RESET COMMAND IS GIVEN
                                        ALL REGISTERS ARE CHECKED

                        2728    ;*******************************************************************

                        2824    ;*******************************************************************
                                TEST 37 READ IN PRESET

                        2827            ALL POSSIBLE REGISTERS WILL BE FILLED WITH ONES
                                        THE REGISTER CONTENTS WILL BE SAVED IN REINTO BUFFER
                                        THE READ IN PRESET COMMAND WILL BE GIVEN
                                        ALL REGISTERS WILL BE CHECKED

                        2832    ;*******************************************************************

                        2907    ;*******************************************************************
                                TEST 40 NO OPERATION FUNCTION TEST
                                        ALL POSSIBLE REGISTERS ARE CLEARED THEN A"NOP"=0
                                        IS GIVEN NO CHANGE SHOULD HAPPEN
                                        ALL POSSIBLE REGISTERS ARE FILLED WITH ONES THEN A "NOP"
                                        IS GIVEN NO CHANGE SHOULD HAPPEN

                        2914    ;*******************************************************************

                        3047    ;*******************************************************************
                                TEST 41 DRIVE CLEAR

                        3050            ALL WRITE BITS OF ALL REGISTERS EXCEPT RHDB ARE FILLED W
                                        ONES EXCEPT FOR BIT #0 AND BIT #6 WHICH ARE "GO" AND
                                        "ENABLE INTERRUPT" BITS
                                        THEN A DRIVE CLEAR IS PERFORMED
                                        THEN ALL REGISTERS EXCEPT RHDB ARE CHECKED
```

```
                                MAINDEC-11-DERPS-B                                DECDOC VER  00.04

                        3056    ;*****************************************************************
                        3331    ;*****************************************************************
                                ;TEST 42 SEEK COMMAND TEST
                        3334    ;       THE SEEK COMMAND WILL BE LOADED INTO RHCS1 WITH GO
                                ;       THEN ALL REGISTERS WILL BE CHECKED
                                ;       RH CLEAR WILL BE GIVEN
                                ;       THEN ALL REGISTERS WILL BE CHECKED
                        3339    ;*****************************************************************
                        3573    ;*****************************************************************
                                ;TEST 43 UNLOAD COMMAND TEST
                        3576    ;       THE UNLOAD COMMAND WILL BE LOADED INTO RHCS1 WITH GO
                                ;       THEN ALL REGISTERS WILL BE CHECKED
                                ;       RH CLEAR WILL BE GIVEN
                        3580    ;*****************************************************************
                        3700    ;*****************************************************************
                                ;TEST 44 OFFSET COMMAND TEST
                        3703    ;       THE OFFSET COMMAND WILL BE LOADED INTO RHCS1 WITH GO
                                ;       THEN ALL REGISTERS WILL BE CHECKED
                                ;       RH CLEAR WILL BE GIVEN
                                ;       THEN ALL REGISTERS WILL BE CHECKED
                        3708    ;*****************************************************************
                        3845    ;*****************************************************************
                                ;TEST 45 RETURN TO CENTER LINE COMMAND TEST
                        3848    ;       THE RETURN TO CENTER LINE COMMAND WILL BE LOADED INTO RH
                                ;       THEN ALL REGISTERS WILL BE CHECKED
                                ;       RH CLEAR WILL BE GIVEN
                                ;       THEN ALL REGISTERS WILL BE CHECKED
                        3853    ;*****************************************************************
                        3981    ;*****************************************************************
                                ;TEST 46 RECALIBRATE COMMAND TEST
                        3984    ;       THE RECALIBRATE COMMAND WILL BE LOADED INTO RHCS1 WITH G
                                ;       THEN ALL REGISTERS WILL BE CHECKED
                                ;       RH CLEAR WILL BE GIVEN
                                ;       THEN ALL REGISTERS WILL BE CHECKED
```

```
                                MAINDEC-11-DERPS-B                                                  DECDOC VER  00.04

                        3989    ;*****************************************************************
                        4116    ;*****************************************************************
                                ;TEST 47 RELEASE COMMAND TEST
                        4119    ;       THE RELEASE COMMAND WILL BE LOADED INTO RHCS1 WITH GO
                                ;       THEN ALL REGISTERS WILL BE CHECKED
                                ;       RH CLEAR WILL BE GIVEN
                                ;       THEN ALL REGISTERS WILL BE CHECKED
                        4124    ;*****************************************************************
                        4191    ;*****************************************************************
                                ;TEST 50 MAKE CURRENT CYLINDER = 0
                        4193    ;*****************************************************************
                        4206    ;*****************************************************************
                                ;TEST 51 LOOK AHEAD REGISTER
                        4209    ;       A SEARCH COMMAND IS GIVEN FOR CYLINDER 0, TRACK 0, SECTO
                                ;       THE LOOK AHEAD REGISTER IS CHECKED AFTER INDEX PULSE
                                ;       THE EXTENSION FIELD IS CHECKED IN EACH SECTOR AFTER
                                ;       128 BYTES THEN AGAIN AFTER 128 MORE BYTES THEN AGAIN AFT
                                ;       THE SECTOR COUNT FIELD IS CHECKED AFTER EACH SECTOR
                                ;       AT THE END ALL REGISTERS ARE CHECKED
                        4216    ;*****************************************************************
                        4419    ;*****************************************************************
                                ;TEST 52 MAKE CURRENT CYLINDER = 0
                                ;*****************************************************************
                        4435    ;*****************************************************************
                                ;TEST 53 WRITE HEADER AND DATA 1
                        4438    ;       WRITE CYLINDER 0, FORMAT 16 BIT PER WORD
                                ;       TRACK 0, SECTOR 0, KEYS 0, NUMBER OF WORDS 256 OF 0'S
                                ;       AS EVERYTHING IS ZERO THIS PROVES VERY LITTLE
                                ;       ANY ERROR LOGIC INDICATIONS ARE NOT CONCLUSIVE ON FIRST
                                ;       BECAUSE ERROR LOGIC HAS NOT BEEN CHECKED YET (ON FIRST P
                        4444    ;*****************************************************************
                        4539    ;*****************************************************************
                                ;TEST 54 WRITE HEADER AND DATA 2
                        4542    ;       WRITE CYLINDER0, FORMAT 16 BITS PER WORD
                                ;       TRACK 0, SECTOR 1, KEYS 0, NUMBER OF WORDS 256
                                ;       OF ALL ONES.
                                ;       ANY ERROR LOGIC INDICATIONS ARE NOT CONCLUSIVE ON FIRST
                                ;       BECAUSE ERROR LOGIC HAS NOT BEEN CHECKED YET (ON FIRST P
```

```
1003                                MAINDEC-11-DERPS-B                                  DECDOC VER  00.04
1004
1005                                4548    ;*****************************************************************
1006
1007                                4656    ;*****************************************************************
1008                                        TEST 55 WRITE HEADER AND DATA 3
1009                                                WRITE CYLINDER 0 FORMAT 16 BITS PER WORD
1010                                                TRACK 1, SECTOR 1, KEY 0, NUMBER OF WORDS 256
1011                                                ALTERNATE ONES AND ZEROS (052525)
1012                                                ANY ERROR LOGIC INDICATIONS ARE NOT CONCLUSIVE ON FIRST
1013                                                BECAUSE ERROR LOGIC HAS NOT BEEN CHECKED YET (ON FIRST P
1014
1015                                4664    ;*****************************************************************
1016
1017                                4775    ;*****************************************************************
1018                                        TEST 56 PROGRAM ERROR RHCS2 #10
1019
1020                                4778            WRITE CYLINDER 0, FORMAT 16 BIT PER WORD
1021                                                TRACK 0, SECTOR 0, KEYS 0, NUMBER OF WORDS 256 OF 0'S
1022                                                WHILE GO BIT IS SET ANOTHER GO IS GIVEN THIS SHOULD SET
1023                                                PROGRAM ERROR
1024
1025                                4783    ;*****************************************************************
1026
1027                                4846    ;*****************************************************************
1028                                                THESE TESTS ARE THROUGH THE MAINTAINABILITY REGISTER
1029                                                THE SECTOR GAP AND SYNC BYTE ARE ALWAYS READ AS
1030                                                ZEROS AND 144000 NO MATTER WHAT IS IN THE SIMULATED DISK
1031                                                TAGED SECGAP: AND WSSYNC:
1032                                                THE HEADER CONSISTING OF CYLINDER ADDRESS, SECTOR/
1033                                                TRACK AND THE KEYS ARE READ FROM LOCATION
1034                                                CYL:, SECTOR:, KEY1:, AND KEY2 AND NOT FROM
1035                                                HEADER: ON SIMULATED DISK
1036                                                CRC IS READ FROM SIMULATED DISK LOCATION WCRC:
1037                                                HEADER GAP IS ALWAYS READ AS ZEROS NO MATTER
1038                                                WHAT IS ON THE SIMULATED DISK AREA
1039                                                THE DATA SYNC IS READ  FROM HDWSYN:
1040                                                ON SIMULATED DISK
1041                                                ALL DATA IS READ FROM SIMULATED DISK DISK:
1042
1043                                4863    ;*****************************************************************
1044
1045                                4868    ;*****************************************************************
1046                                        TEST 57 READ HEADER AND DATA 1
1047                                                READ CYLINDER 0 FORMAT 16 BITS PER WORD
1048                                                TRACK 0, SECTOR 0, KEYS 0, 256 WORDS OF 0
1049                                                ANY ERROR LOGIC INDICATIONS ARE NOT CONCLUSIVE ON FIRST
1050                                                BECAUSE ERROR LOGIC HAS NOT BEEN CHECKED YET (ON FIRST P
1051
1052                                4875    ;*****************************************************************
1053
```

```
1054                                MAINDEC-11-DERPS-B                                  DECDOC VER  00.04
1055
1056                                        4990    ;*****************************************************************
1057                                                ;TEST 60 READ HEADER AND DATA 2
1058                                                ;       READ CYLINDER 0 FORMAT 16 BITS PER WORD
1059                                                ;       TRACK 0, SECTOR 1, KEYS 0, 256 WORDS OF 177777
1060                                                ;       ANY ERROR LOGIC INDICATIONS ARE NOT CONCLUSIVE ON FIRST
1061                                                ;       BECAUSE ERROR LOGIC HAS NOT BEEN CHECKED YET (ON FIRST P
1062
1063                                        4997    ;*****************************************************************
1064
1065                                        5111    ;*****************************************************************
1066                                                ;TEST 61 READ HEADER AND DATA 3
1067                                                ;       READ CYLINDER 0 FORMAT 16 BITS PER WORD
1068                                                ;       TRACK 1, SECTOR 1, KEYS 0, 256 WORDS OF 052525
1069                                                ;       ANY ERROR LOGIC INDICATIONS ARE NOT CONCLUSIVE ON FIRST
1070                                                ;       BECAUSE ERROR LOGIC HAS NOT BEEN CHECKED YET (ON FIRST P
1071
1072                                        5118    ;*****************************************************************
1073
1074                                        5232    ;*****************************************************************
1075                                                ;TEST 62 WRITE DATA
1076
1077                                        5235    ;       WRITE CYLINDER 0, FORMAT 16 BITS PER WORD
1078                                                ;       TRACK 0, SECTOR 0, KEYS 0, NUMBER OF WORDS 256 OF 377
1079                                                ;       ANY ERROR LOGIC INDICATION IS NOT CONCLUSIVE ON FIRST PA
1080                                                ;       BECAUSE ERROR LOGIC HAS NOT BEEN CHECKED
1081
1082                                        5240    ;*****************************************************************
1083
1084                                        5322    ;*****************************************************************
1085                                                ;TEST 63 READ DATA
1086
1087                                        5325    ;       READ CYLINDER0, FORMAT 16 BITS PER WORD
1088                                                ;       TRACK0, SECTOR 1, KEYS 0, 10 WORDS OF 177400
1089
1090                                        5327    ;       ANY ERROR LOGIC INDICATION IS NOT CONCLUSIVE
1091                                                ;       BECAUSE ERROR LOGIC HAS NOT BEEN CHECKED YET (ON FIRST P
1092
1093                                        5330    ;*****************************************************************
1094
1095                                        5440    ;*****************************************************************
1096                                                ;TEST 64 WRITE CHECK HEADER AND DATA
1097
1098                                        5443    ;       WRITE CHECK CYLINDER 0, FORMAT 16 BITS PER WORD
1099                                                ;       TRACK 1, SECTOR 1, KEYS 0, 36 WORDS AS SHOWN BELOW
1100                                                ;       ANY DEVICE LOGIC ERROR INDICATION IS NOT CONCLUSIVE ON F
1101                                                ;       BECAUSE ERROR LOGIC HAS NOT BEEN CHECKED YET (ON FIRST P
1102                                                ;       ONLY RH WRITE CHECK ERROR (RHCS2 BIT 14) IS TESTED HERE
1103                                                ;*****************************************************************
1104
```

```
1105                            MAINDEC-11-DERPS-B                                  DECDOC VER  00.04
1106
1107                                    5574    ;*****************************************************************
1108                                            TEST 65 WRITE CHECK DATA
1109
1110                                    5577            WRITE CHECK DATA CYLINDER 0 FORMAT 16 BITS PER WORD
1111                                                    TRACK 1, SECTOR 1, KEYS 0, 32 WORDS OF DATA
1112                                                    ANY DEVICE LOGIC ERROR INDICATIONS ARE NOT CONCLUSIVE ON
1113                                                    BECAUSE ERROR LOGIC HAS NOT YET (ON FIRST PASS) BEEN CHE
1114                                                    ONLY RH WRITE CHECK ERROR IS TESTED
1115
1116                                    5583    ;*****************************************************************
1117
1118                                    5708    ;*****************************************************************
1119                                            TEST 66 ATTENTION WITH ERROR TEST
1120
1121                                    5711            THIS TESTS THE SETTING OF ATA BIT BOTH IN THE RHAS
1122                                                    AND THE RHDS1 REGISTERS WITH THE SETTING OF EACH
1123                                                    ERROR BIT ON THE THREE ERROR REGISTERS.
1124                                                    IN EACH OF THE ABOVE CASES ERR IN RHDS1 SHOULD
1125                                                    ALSO SET
1126                                                    "GO" SHOULD CLEAR ERR, ATA IN RHDS1 AND RHAS BUT NOT ERR
1127                                                    PUTTING "1" IN RHAS DRIVE POSITION CLEARS DRIVE BIT IN A
1128                                                    UPPER BYTE OF RHAS IS INVALIBD
1129
1130                                    5720    ;*****************************************************************
1131
1132                                    5829    ;*****************************************************************
1133                                            TEST 67 BUS ADDRESS INHIBIT
1134
1135                                    5832            READ CYLINDER0, FORMAT 16 BITS PER WORD
1136                                                    TRACK0, SECTOR 1, KEYS 0, 10 WORDS OF 177400
1137                                                    THIS IS DONE WITH BUS ADDRESS INHIBIT SET
1138                                                    ANY ERROR LOGIC INDICATION IS NOT CONCLUSIVE ON FIRST PA
1139                                                    BECAUSE ERROR LOGIC HAS NOT BEEN CHECKED YET (ON FIRST P
1140
1141                                    5838    ;*****************************************************************
1142
1143                                    5944    ;*****************************************************************
1144                                            TEST 70 RHCS2 - BIT # 11 - NEM
1145
1146                                    5947            READ CYLINDER0, FORMAT 16 BITS PER WORD
1147                                                    TRACK0, SECTOR 1, KEYS 0, 1 WORD OF 177400
1148                                                    THIS IS DONE WITH BUS ADDRESS INHIBIT SET
1149                                                    BUS ADDRESS USED IS 760000 THIS IS ALWAYS NON EXISTANT
1150                                                    THIS SHOULD SET NEM
1151
1152                                    5953    ;*****************************************************************
1153
1154                                    6036    ;*****************************************************************
1155                                            TEST 71 WRITE CHECK ERROR
1156
```

```
                                MAINDEC-11-DERPS-B                                              DECDOC VER  00.04

                        6039            WRITE CHECK DATA CYLINDER 0 FORMAT 16 BITS PER WORD
                                        TRACK 1, SECTOR 1, KEYS 0, 32 WORDS OF DATA
                                        FIFTH WORD IS CHANGED ON DISK TO GIVE WRITE CHECK ERROR
                                        ANY DEVICE LOGIC ERROR INDICATIONS ARE NOT CONCLUSIVE
                                        ON FIRST PASS
                                        BECAUSE ERROR LOGIC HAS NOT YET BEEN CHECKED
                                        ONLY RH WRITE CHECK ERROR IS TESTED

                        6047    ;*******************************************************************

                        6187    ;*******************************************************************
                                TEST 72 ERROR REGISTER #1-BIT 4 -FORMAT ERROR
                                        THE SIMULATED DISK IS FILLED WITH CYLINGER 0 TRACK 1
                                        SECTOR 0 FORMAT=18 BITS PER WORD AND 4 WORDS

                        6191            OF 125252, A READ HEADER AND DATA COMMAND IS GIVEN WITH
                                        PER WORD FORMAT, FER=BIT4 SHOULD SET BUT THE
                                        READ SHOULD BE COMPLETE

                        6195    ;*******************************************************************

                        6325    ;*******************************************************************
                                TEST 73 ERROR REGISTER #1-BIT 4 -FORMAT ERROR

                        6328            THE SIMULATED DISK HEADER IS FILLED WITH CYLINDER 0
                                        TRACK 0, SECTOR 0 FORMAT 18 BITS PER WORD
                                        A WRITE DATA COMMAND IS GIVEN WITH SAME HEADER
                                        EXCEPT FORMAT BIT.  THE DATA SHOULD NOT BE WRITTEN.

                        6333    ;*******************************************************************

                        6427            ERROR REGISTER #01 (RHER1) TEST
                                        BIT #1 (ILLEGAL REGISTER) CANNOT BE TESTED ON PDP11 THIS
                                        IS FOR PDP10 USE ONLY

                        6432    ;*******************************************************************
                                TEST 74 TEST ILF BIT #0 IN REG. RHER1

                        6435            ILLEGAL FUNCTION SHOULD SET ATA,ERR,ILF
                                        A GO WITHOUT CLEARING ERR SHOULD SET MXF,DLT,TRE

                        6438    ;*******************************************************************

                        6528    ;*******************************************************************
                                TEST 75 RHER1- BIT #2 - REG. MODIFICATION REFUSED

                        6531            IN THIS TEST THE REGISTERS ARE IN TWO GROUPS
                                        FIRST - RHCS1,RHDST,RHOF,RHCA,RHER1,RHER2,RHER3 - SETS R
                                        SECOND - RHMR,RHAS - DOES NOT SET RMR
                                        IF WRITING IS ATTEMPTED DURING AN OPERATION


                                        ONLY ONE REGISTER IS WRITTEN INTO THAT IS RHCA
```

```
1213
```

```
                                        1 THE REGISTERS CONTENTS ARE SAVED IN "REINTO" BUFFER
                                        2 WRITE HEADER AND DATA IS STARTED
                                        3 ATTEMPT IS MADE TO WRITE INTO REGISTERS
                                        4 ALL REGISTERS ARE COMPARED

                                6545    ;*****************************************************************
                                6631    ;*****************************************************************
                                        TEST 76 MAKE CURRENT CYLINDER = 1
                                        ;*****************************************************************
                                6646    ;*****************************************************************
                                        TEST 77 ERROR REG1 - BIT #7 - HEADER COMPARE ERROR
                                6649            THE SIMULATED DISK IS SET TO READ CYLINDER=0, TRACK=1
                                                SECTOR=1, KEYS=1, 256 WORDS OF 177400
                                                A READ HEADER AND DATA COMMAND IS GIVEN TO READ
                                                CYLINDER=1, TRACK=1, SECTOR=1, KEY1=1, KEY2=1
                                                REINTO BUFFER IS FILLED WITH 0
                                                WRFROM IS FILLED WITH 10000,401,1,1,1, AND ALL 177400
                                                AFTER THE READ THE REINTO BUFFER IS EXPECTED TO
                                                HAVE WHAT IS IN WRFROM - 10000,401,1,1 AND ALL 177400
                                6658    ;*****************************************************************
                                6715    ;*****************************************************************
                                        TEST 100        MAKE CURRENT CYLINDER = 0
                                        ;*****************************************************************
                                6731    ;*****************************************************************
                                        TEST 101        ERROR REG1 - BIT #7 - HEADER COMPARE ERROR
                                6734            THE SIMULATED DISK IS SET TO READ CYLINDER=0, TRACK=1
                                                SECTOR=1, KEYS=1, 256 WORDS OF 177400
                                                A READ HEADER AND DATA COMMAND IS GIVEN TO READ
                                                CYLINDER=0, TRACK=0, SECTOR=1, KEY1=1, KEY2=1
                                                REINTO BUFFER IS FILLED WITH 0
                                                WRFROM IS FILLED WITH 10000,401,1,1,1, AND ALL 177400
                                                AFTER THE READ THE REINTO BUFFER IS EXPECTED TO
                                                HAVE WHAT IS IN WRFROM - 10000,401,1,1 AND ALL 177400
                                6743    ;*****************************************************************
                                6798    ;*****************************************************************
                                        TEST 102        MAKE CURRENT CYLINDER = 1
                                        ;*****************************************************************
                                6816    ;*****************************************************************
                                        TEST 103        ERROR REG.1 - BIT #7 - HEADER COMPARE ERROR
```

```
                                MAINDEC-11-DERPS-B                                                DECDOC VER  00.04

                                                6819            THE SIMULATED DISK IS SET UP FOR CYLINDER=0, TRACK=1
                                                                SECTOR=1, KEYS=1, 256 WORDS OF 177400
                                                                A WRITE DATA COMMAND IS GIVEN TO WRITE CYLINDER=1
                                                                TRACK=1, SECTOR=1, KEY1=1, KEY2=1
                                                                WRFROM BUFFER IS FILLED WITH 125252
                                                                REINTO BUFFER IS FILLED WITH 177400
                                                                AFTER THE WRITE COMMAND THE DISK IS EXPECTED TO
                                                                HAVE 177400

                                                6828    ********************************************************************

                                                6882    ********************************************************************
                                                        TEST 104        MAKE CURRENT CYLINDER = 0
                                                        ********************************************************************

                                                6897    ********************************************************************
                                                        TEST 105        ERROR REG.1 - BIT #7 - HEADER COMPARE ERROR

                                                6900            THE SIMULATED DISK IS SET UP FOR CYLINDER=0, TRACK=1
                                                                SECTOR=1, KEYS=1, 256 WORDS OF 177400
                                                                A WRITE DATA COMMAND IS GIVEN TO WRITE CYLINDER=0
                                                                TRACK=0, SECTOR=1, KEY1=1, KEY2=1
                                                                WRFROM BUFFER IS FILLED WITH 125252
                                                                REINTO BUFFER IS FILLED WITH 177400
                                                                AFTER THE WRITE COMMAND THE DISK IS EXPECTED TO
                                                                HAVE 177400

                                                6909    ********************************************************************

                                                6960    ********************************************************************
                                                        TEST 106        ERROR REG.1 - BIT #8 - CRC ERROR

                                                6963            THE SIMULATED DISK IS SET TO READ CYLINDER=0, TRACK=1
                                                                SECTOR=1, KEYS=1, 256 WORDS OF 177400
                                                                A READ HEADER AND DATA COMMAND IS GIVEN TO READ
                                                                CYLINDER=0, TRACK=1, SECTOR=1, KEY1=1, KEY2=1
                                                                REINTO BUFFER IS FILLED WITH 0
                                                                WRFROM IS FILLED WITH 10000,401,1,1,1, AND ALL 177400
                                                                AFTER THE READ THE REINTO BUFFER IS EXPECTED TO
                                                                HAVE WHAT IS IN WRFROM - 10000,401,1,1 AND ALL 177400

                                                6972    ********************************************************************

                                                7028    ********************************************************************
                                                        TEST 107        ERROR REG.1 - BIT #8 - CRC ERROR

                                                7031            THE SIMULATED DISK IS SET TO READ CYLINDER=0, TRACK=1
                                                                SECTOR=1, KEYS=1, 256 WORDS OF 177400
                                                                A READ HEADER AND DATA COMMAND IS GIVEN TO READ
                                                                CYLINDER=0, TRACK=1, SECTOR=1, KEY1=1, KEY2=1
                                                                REINTO BUFFER IS FILLED WITH 0
                                                                WRFROM IS FILLED WITH 10000,401,1,1,1, AND ALL 177400
                                                                AFTER THE READ THE REINTO BUFFER IS EXPECTED TO
                                                                HAVE WHAT IS IN WRFROM - 10000,401,1,1 AND ALL 177400
```

```
1323
```

```
1324                            MAINDEC-11-DERPS-B                                      DECDOC VER  00.04
1325
1326                                    7040    ;*****************************************************************
1327
1328                                    7096    ;*****************************************************************
1329                                            TEST 110        ERROR REG.1 - BIT 6 - CRC ERROR
1330
1331                                    7099            THE SIMULATED DISK IS SET UP FOR CYLINDER=0, TRACK=1
1332                                            SECTOR=1, KEYS=1, 256 WORDS OF 177400
1333                                            A WRITE DATA COMMAND IS GIVEN TO WRITE CYLINDER=0
1334                                            TRACK=1, SECTOR=1, KEY1=1, KEY2=1
1335                                            WRFROM BUFFER IS FILLED WITH 125252
1336                                            REINTO BUFFER IS FILLED WITH 177400
1337                                            AFTER THE WRITE COMMAND THE DISK IS EXPECTED TO
1338                                            HAVE 177400
1339
1340                                    7108    ;*****************************************************************
1341
1342                                    7161    ;*****************************************************************
1343                                            TEST 111        ERROR REG.1 - BIT #8 - CRC ERROR
1344
1345                                    7164            THE SIMULATED DISK IS SET UP FOR CYLINDER=0, TRACK=1
1346                                            SECTOR=1, KEYS=1, 256 WORDS OF 177400
1347                                            A WRITE DATA COMMAND IS GIVEN TO WRITE CYLINDER=0
1348                                            TRACK=1, SECTOR=1, KEY1=1, KEY2=1
1349                                            WRFROM BUFFER IS FILLED WITH 125252
1350                                            REINTO BUFFER IS FILLED WITH 177400
1351                                            AFTER THE WRITE COMMAND THE DISK IS EXPECTED TO
1352                                            HAVE 177400
1353
1354                                    7173    ;*****************************************************************
1355
1356                                    7229    ;*****************************************************************
1357                                            TEST 112        MAKE CURRENT CYLINDER = 410.
1358                                            ;*****************************************************************
1359
1360                                    7246    ;*****************************************************************
1361                                            TEST 113        RHDS1 - BIT #10 LAST SECTOR TRANSFERRED
1362                                                    WRITE CYLINDER 410., FORMAT 16 BITS PER WORD
1363                                                    TRACK 18., SECTOR 21., KEYS 0, NUMBER OF WORDS
1364                                                    256., OF 377
1365                                                    LST BIT # 10 RHDS1 SHOULD SET AFTER WRITE
1366                                                    IS COMPLETE.
1367
1368                                    7254    ;*****************************************************************
1369
1370                                    7348    ;*****************************************************************
1371                                            TEST 114        ERROR REGISTER 1 - BIT #9 AOE
1372
1373                                    7351            A WRITE DATA COMMAND IS GIVEN TO CYLINDER 410
1374                                            SECTOR 21 TRACK 18, KEYS 0, DATA 377
1375                                            WORD COUNT REGISTER FOR 326 (256+66+4) WORDS
1376
1377                                            AFTER 256 WORDS HAVE BEEN WRITTEN
1378                                            AOE SHOULD COME UP
1379                                            RHWC WILL SHOW 4 BECAUSE THE SILO IS 66 WORDS AND
```

```
1380
```

```
                                        256 WORDS HAVE BEEN WRITTEN - TOTAL 322
                                        THIS IS 4 SHORT OF 326

                7361    ;*****************************************************************
                7490    ;*****************************************************************
                        ;TEST 115       MAKE CURRENT CYLINDER = 0
                        ;*****************************************************************

                7507    ;*****************************************************************
                        ;TEST 116       ERROR REGISTER 1- BIT #10 IAE

                7510            A READ HEADER AND DATA IS GIVEN TO TRACK 20
                                SECTOR 0

                                AN INDEX PULSE IS GIVEN TO GET RHLA TO 0

                                IAE BIT SHOULD SET

                7517    ;*****************************************************************
                7613    ;*****************************************************************
                        ;TEST 117       ERROR REGISTER 1- BIT #10 IAE

                7616            A WRITE HEADER AND DATA IS GIVEN TO SECTOR 22
                                TRACK 0 CYLINDER 0

                                WORD COUNT IS SET TO 256.

                                AN INDEX PULSE IS GIVEN TO GET RHLA TO 0

                                IAE BIT SHOULD SET

                7625    ;*****************************************************************
                7726    ;*****************************************************************
                        ;TEST 120       ERROR REGISTER 1- BIT #10 IAE

                7729            A WRITE DATA IS GIVEN TO SECTOR 0
                                TRACK 0 CYLINDER 411

                                WORD COUNT IS SET TO 256.
                                AN INDEX PULSE IS GIVEN TO GET RHLA TO 0


                                IAE BIT SHOULD SET

                7738    ;*****************************************************************
```

```
                        7836    ;*****************************************************************
                                ;*****************************************************************
                                TEST 121        END OF DRIVE

                        7840            THIS IS THE END OF TEST FOR ONE DRIVE
                                        IF THERE ARE MORE DRIVES THEN THE PROGRAM
                                        JUMPS TO TEST 5 FOR NEXT DRIVE TEST
                                        END PASS IS REACHED ONLY AFTER ALL DRIVES ARE COMPLETE

                        7845    ;*****************************************************************

                        7880    ;*****************************************************************

                                ;*****************************************************************
                7882    END OF PASS ROUTINE
                                ;*****************************************************************

                        7884    INCREMENT THE PASS NUMBER ($PASS)
                                TYPE "END PASS #XXXXX" (WHERE XXXXX IS A DECIMAL NUMBER)
                                IF THERES A MONITOR GO TO IT
                                IF THERE ISN'T JUMP TO TST1

                                ;*****************************************************************
                7924    SUBROUTINES
                                ;*****************************************************************

                                ;*****************************************************************
                7990    SAVE REGISTERS ROUTINE
                                ;*****************************************************************

                                ;*****************************************************************
                8018    FLOAT 1 AND 0
                                ;*****************************************************************

                                ;*****************************************************************
                8073    CLEAR MEMORY ROUTINE
                                ;*****************************************************************

                                ;*****************************************************************
                8106    LOCAL TRAPS
                                ;*****************************************************************
```

```
                        **********************************************************************
                8123    CLEAD DISK ROUTINE
                        **********************************************************************

                        **********************************************************************
                8136    CHECK DISK STATUS ROUTINE
                        **********************************************************************

                        **********************************************************************
                8263    SAVE ROUTINE
                        **********************************************************************

                        **********************************************************************
                8288    WRITE CHECK ROUTINE
                        **********************************************************************

                        **********************************************************************
                8324    COMPARE ROUTINE
                        **********************************************************************

                        **********************************************************************
                8414    CRC GENERATION ROUTINE
                        **********************************************************************

                        **********************************************************************
                8727    JAM CURRENT CYLINDER ROUTINE
                        **********************************************************************

                        **********************************************************************
                8764    ECC GENERARION AND COMPARISON ROUTINE
                        **********************************************************************

                        **********************************************************************
                9100    RH BASE ADDRESS CHANGE ROUTINE
                        **********************************************************************

                9101            THIS ROUTINE WILL ALLOW THE CHANGE OF THE BASE
                                ADDRESS FROM 176700 TO ANY TYPED VALUE
```

```
1517
1518
1519                                    **********************************************************************
1520                            9172    DISK SIMULATION
1521                                    **********************************************************************
1522
1523                                    9173    **********************************************************************
1524                                            **********************************************************************
1525
1526                                    9319    **********************************************************************
1527
1528                                    9341    **********************************************************************
1529
1530                                    9348    **********************************************************************
1531                                            **********************************************************************
1532
1533                                    9475    **********************************************************************
1534                                            **********************************************************************
1535
1536                                    9542    **********************************************************************
1537                                            **********************************************************************
1538
1539                                    9610    **********************************************************************
1540                                            **********************************************************************
1541
1542                                    9683    **********************************************************************
1543                                            **********************************************************************
1544
1545                                    9715    **********************************************************************
1546                                            **********************************************************************
1547
1548                                    9832    **********************************************************************
1549                                            **********************************************************************
1550
1551                                    9976    **********************************************************************
1552                                            **********************************************************************
1553
1554                                    10089   **********************************************************************
1555
1556                                    10091   **********************************************************************
1557
1558                                    10138   .**************************
1559                                            **********************************************************************
1560
```

```
                                MAINDEC-11-DERPS-B                                              DECDOC VER  00.04

                                ;*********************************************************************
10141                           ;SCOPE HANDLER ROUTINE
                                ;*********************************************************************

                                10143   THIS ROUTINE CONTROLS THE LOOPING OF SUBTESTS. IT WILL INCREMENT
                                        AND LOAD THE TEST NUMBER($TSTNM) INTO THE DISPLAY REG.(DISPLAY<7
                                        AND LOAD THE ERROR FLAG ($ERFLG) INTO DISPLAY<15:08>
                                        THE SWITCH OPTIONS PROVIDED BY THIS ROUTINE ARE:
                                        SW14=1  LOOP ON TEST
                                        SW11=1  INHIBIT ITERATIONS
                                        SW09=1  LOOP ON ERROR
                                        SW08=1  LOOP ON TEST IN SWR<7:0>
                                        CALL
                                                SCOPE           ;;SCOPE=IOT

                                10213   ;*********************************************************************

                                ;*********************************************************************
10215                           ;CONVERT BINARY TO DECIMAL AND TYPE ROUTINE
                                ;*********************************************************************

                                10217   THIS ROUTINE IS USED TO CHANGE A 16-BIT BINARY NUMBER TO A 5-DIG
                                        SIGNED DECIMAL (ASCII) NUMBER AND TYPE IT. DEPENDING ON WHETHER
                                        NUMBER IS POSITIVE OR NEGATIVE A SPACE OR A MINUS SIGN WILL BE T
                                        BEFORE THE FIRST DIGIT OF THE NUMBER. LEADING ZEROS WILL ALWAYS
                                        REPLACED WITH SPACES.
                                        CALL:
                                                MOV     NUM,-(SP)       ;;PUT THE BINARY NUMBER ON THE S
                                                TYPDS                   ;;GO TO THE ROUTINE

                                10281   ;*********************************************************************

                                ;*********************************************************************
10283                           ;TYPE ROUTINE
                                ;*********************************************************************

                                10285   ROUTINE TO TYPE ASCIZ MESSAGE. MESSAGE MUST TERMINATE WITH A 0 B
                                        THE ROUTINE WILL INSERT A NUMBER OF NULL CHARACTERS AFTER A LINE
                                        NOTE1:  $NULL CONTAINS THE CHARACTER TO BE USED AS THE FILLER CH
                                        NOTE2:  $FILLS CONTAINS THE NUMBER OF FILLER CHARACTERS REQUIRED
                                        NOTE3:  $FILLC CONTAINS THE CHARACTER TO FILL AFTER.

                                        CALL:
                                        1) USING A TRAP INSTRUCTION
                                                TYPE    ,MESADR         ;;MESADR IS FIRST ADDRESS OF AN
                                        OR
                                                TYPE
                                                MESADR

                                        2) USING A JSR INSTRUCTION
                                                MOV     PS,-(SP)        ;;PUSH PROCESSOR STATUS WORD ON
                                                JSR     PC,$TYPE        ;;CALL TYPE ROUTINE
                                                MESADDR                 ;;FIRST ADRESS OF MESSAGE
```

```
1617                            MAINDEC-11-DERPS-B                                  DECDOC VER  00.04
1618
1619                                    10357   ;*******************************************************************
1620
1621                                    ;*******************************************************************
1622                            10359   ;TTY INPUT ROUTINE
1623                                    ;*******************************************************************
1624
1625                                    10368   ;TK INITIALIZE ROUTINE
1626                                            ;THIS ROUTINE WILL INITIALIZE THE TTY KEYBOARD INPUT QUEUE
1627                                            ;SETUP THE INTERRUPT VECTOR AND TURN ON THE KEYBOARD INTERRUPT
1628                                            ;CALL:
1629                                                    JSR     PC,$TKINT
1630                                                    RETURN
1631
1632                                    10385   ;TK SERVICE ROUTINE
1633                                            ;THIS ROUTINE WILL SERVICE THE TTY KEYBOARD INTERRUPT
1634                                            ;BY READING THE CHARACTER FROM THE INPUT BUFFER AND PUTTING
1635                                            ;IT IN THE QUEUE.
1636                                            ;IF THE CHARACTER IS A "CONTROL-C" (↑C) $TKINT IS CALLED AND
1637                                            ;UPON RETURN EXIT IS MAKE TO THE "CONTROL-C" RESTART ADDRESS (OPE
1638
1639                                    10411   ;*******************************************************************
1640                                            ;THIS ROUTINE WILL INPUT A SINGLE CHARACTER FROM THE TTY
1641                                            ;CALL:
1642                                                    RDCHR                   ;;GET A CHARACTER FROM THE QUEUE
1643                                                    RETURN HERE             ;;CHARACTER IS ON THE STACK
1644                                                                            ;;WITH PARITY BIT STRIPPED OFF
1645
1646                                    10432   ;*******************************************************************
1647                                            ;THIS ROUTINE WILL INPUT A STRING FROM THE TTY
1648                                            ;CALL:
1649                                                    RDLIN                   ;;INPUT A STRING FROM THE TTY
1650                                                    RETURN HERE             ;;ADDRESS OF FIRST CHARACTER WIL
1651                                                                            ;;TERMINATOR WILL BE A BYTE OF A
1652
1653                                    10468   ;*******************************************************************
1654
1655                                    ;*******************************************************************
1656                            10470   ;READ AN OCTAL NUMBER FROM THE TTY
1657                                    ;*******************************************************************
1658
1659                                    10472   ;THIS ROUTINE WILL READ AN OCTAL (ASCII) NUMBER FROM THE TTY AND
1660                                            ;CHANGE IT TO BINARY.
1661                                            ;THE INPUT CHARACTERS WILL BE CHECKED TO INSURED THEY ARE LEGAL
1662                                            ;OCTAL DIGITS. IF AN ILLEGAL CHARACTER IS READ A "?" WILL BE TYPE
1663                                            ;FOLLOWED BY A CARRIAGE RETURN-LINE FEED. THE COMPLETE NUMBER MUS
1664                                            ;THEN BE RETYPED. THE INPUT IS TERMINATED BY TYPING A CARRIAGE RE
1665                                            ;CALL:
1666                                                    RDOCT                   ;;READ AN OCTAL NUMBER
1667                                                    RETURN HERE             ;;LOW ORDER BITS ARE ON TOP OF T
1668                                                                            ;;HIGH ORDER BITS ARE IN $HIOCT
1669
```

```
                        10522   ;*****************************************************************
                                ;*****************************************************************
                10524   ERROR HANDLER ROUTINE
                                ;*****************************************************************
                        10526   THIS ROUTINE WILL INCREMENT THE ERROR FLAG AND THE ERROR COUNT,
                                SAVE THE ERROR ITEM NUMBER AND THE ADDRESS OF THE ERROR CALL
                                AND GO TO $ERRTYP ON ERROR
                                THE SWITCH OPTIONS PROVIDED BY THIS ROUTINE ARE:
                                SW15=1  HALT ON ERROR
                                SW13=1  INHIBIT ERROR TYPEOUTS
                                SW10=1  BELL ON ERROR
                                SW09=1  LOOP ON ERROR
                                CALL
                                        ERROR   N       ;;ERROR=EMT AND N=ERROR ITEM NUMBER
                        10568   ;*****************************************************************
                                ;*****************************************************************
                10570   ERROR MESSAGE TYPEOUT ROUTINE
                                ;*****************************************************************
                        10572   THIS ROUTINE USES THE "ITEM CONTROL BYTE" ($ITEMB) TO DETERMINE
                                ERROR IS TO BE REPORTED. IT THEN OBTAINS, FROM THE "ERROR TABLE"
                                AND REPORTS THE APPROPRIATE INFORMATION CONCERNING THE ERROR.
                        10625   ;*****************************************************************
                                ;*****************************************************************
                10628   BINARY TO OCTAL (ASCII) AND TYPE
                                ;*****************************************************************
                        10630   THIS ROUTINE IS USED TO CHANGE A 16-BIT BINARY NUMBER TO A 6-DIG
                                OCTAL (ASCII) NUMBER AND TYPE IT.
                                $TYPOS---ENTER HERE TO SETUP SUPPRESS ZEROS AND NUMBER OF DIGITS
                                CALL:
                                        MOV     NUM,-(SP)       ;;NUMBER TO BE TYPED
                                        TYPOS                   ;;CALL FOR TYPEOUT
                                        .BYTE   N               ;;N=1 TO 6 FOR NUMBER OF DIGITS
                                        .BYTE   M               ;;M=1 OR 0
                                                                ;;1=TYPE LEADING ZEROS
                                                                ;;0=SUPPRESS LEADING ZER

                                $TYPON----ENTER HERE TO TYPE OUT WITH THE SAME PARAMETERS AS THE
                                $TYPOS OR $TYPOC
                                CALL:
                                        MOV     NUM,-(SP)       ;;NUMBER TO BE TYPED
                                        TYPON                   ;;CALL FOR TYPEOUT

                                $TYPOC---ENTER HERE FOR TYPEOUT OF A 16 BIT NUMBER
                                CALL:
                                        MOV     NUM,-(SP)       ;;NUMBER TO BE TYPED
```

```
1726
```

```
                                TYPOC           ;;CALL FOR TYPEOUT

                        10704   ;*****************************************************************

                                ;*****************************************************************
                        10706   ;TRAP DECODER
                                ;*****************************************************************

                        10708   ;THIS ROUTINE WILL PICKUP THE LOWER BYTE OF THE "TRAP" INSTRUCTIO
                                ;AND USE IT TO INDEX THROUGH THE TRAP TABLE FOR THE STARTING ADDR
                                ;OF THE DESIRED ROUTINE. THEN USING THE ADDRESS OBTAINED IT WILL
                                ;GO TO THAT ROUTINE.

                                ;*****************************************************************
                        10721   ;TRAP TABLE
                                ;*****************************************************************

                        10723   ;THIS TABLE CONTAINS THE STARTING ADDRESSES OF THE ROUTINES CALLE
                                ;BY THE "TRAP" INSTRUCTION.

                        10743   ;*****************************************************************

                                ;*****************************************************************
                        10745   ;POWER DOWN AND UP ROUTINES
                                ;*****************************************************************

                        10785   ;*****************************************************************
                                ;*****************************************************************
```

```
                                        %
                                .TITLE  MAINDEC-11-DERPS-B
                                ;*COPYRIGHT (C) 1975
                                ;*DIGITAL EQUIPMENT CORP.
                                ;*MAYNARD, MASS. 01754
                                ;*
                                ;*PROGRAM BY SUB MALLICK
                                ;*
                                ;*THIS PROGRAM WAS ASSEMBLED USING THE PDP-11 MAINDEC SYSMAC
                                ;*PACKAGE (MAINDEC-11-D?QAC-CD),MAR 21, 1976.
                                ;*

                                .SBTTL  OPERATIONAL SWITCH SETTINGS
                                ;*
                                ;*      SWITCH                  USE
                                ;*      ------          --------------------
                                ;*        15            HALT ON ERROR
                                ;*        14            LOOP ON TEST
                                ;*        13            INHIBIT ERROR TYPEOUTS
                                ;*        11            INHIBIT ITERATIONS
                                ;*        10            BELL ON ERROR
                                ;*         9            LOOP ON ERROR
                                ;*         8            LOOP ON TEST IN SWR<7:0>
                                ;*         7            STOP FURTHER COMPARES IF SW08 IS LOW
                                ;*         6            ECC TEST-COMPARE END RESULTS ONLY IF SW08 IS LOW
                                .SBTTL  BASIC DEFINITIONS

                                ;*INITIAL ADDRESS OF THE STACK POINTER *** 1000 ***
001000                          STACK=  1000
                                .EQUIV  EMT,ERROR       ;;BASIC DEFINITION OF ERROR CALL
                                .EQUIV  IOT,SCOPE       ;;BASIC DEFINITION OF SCOPE CALL

                                ;*MISCELLANEOUS DEFINITIONS
000011                          HT=     11              ;;CODE FOR HORIZONTAL TAB
000012                          LF=     12              ;;CODE FOR LINE FEED
000015                          CR=     15              ;;CODE FOR CARRIAGE RETURN
000200                          CRLF=   200             ;;CODE FOR CARRIAGE RETURN-LINE FEED
177776                          PS=     177776          ;;PROCESSOR STATUS WORD
                                .EQUIV  PS,PSW
177774                          STKLMT= 177774          ;;STACK LIMIT REGISTER
177772                          PIRQ=   177772          ;;PROGRAM INTERRUPT REQUEST REGISTER
177570                          DSWR=   177570          ;;HARDWARE SWITCH REGISTER
177570                          DDISP=  177570          ;;HARDWARE DISPLAY REGISTER

                                ;*GENERAL PURPOSE REGISTER DEFINITIONS
000000                          R0=     %0              ;;GENERAL REGISTER
000001                          R1=     %1              ;;GENERAL REGISTER
000002                          R2=     %2              ;;GENERAL REGISTER
000003                          R3=     %3              ;;GENERAL REGISTER
000004                          R4=     %4              ;;GENERAL REGISTER
000005                          R5=     %5              ;;GENERAL REGISTER
000006                          R6=     %6              ;;GENERAL REGISTER
000007                          R7=     %7              ;;GENERAL REGISTER
                                .EQUIV  R6,SP           ;;STACK POINTER
                                .EQUIV  R7,PC           ;;PROGRAM COUNTER
```

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

                                ;*"SWITCH REGISTER" SWITCH DEFINITIONS
100000                          SW15=   100000
040000                          SW14=   40000
020000                          SW13=   20000
010000                          SW12=   10000
004000                          SW11=   4000
002000                          SW10=   2000
001000                          SW09=   1000
000400                          SW08=   400
000200                          SW07=   200
000100                          SW06=   100
000040                          SW05=   40
000020                          SW04=   20
000010                          SW03=   10
000004                          SW02=   4
000002                          SW01=   2
000001                          SW00=   1
                                .EQUIV  SW09,SW9
                                .EQUIV  SW08,SW8
                                .EQUIV  SW07,SW7
                                .EQUIV  SW06,SW6
                                .EQUIV  SW05,SW5
                                .EQUIV  SW04,SW4
                                .EQUIV  SW03,SW3
                                .EQUIV  SW02,SW2
                                .EQUIV  SW01,SW1
                                .EQUIV  SW00,SW0

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

```
                                .EQUIV  BIT08,BIT8
                                .EQUIV  BIT07,BIT7
                                .EQUIV  BIT06,BIT6
                                .EQUIV  BIT05,BIT5
                                .EQUIV  BIT04,BIT4
                                .EQUIV  BIT03,BIT3
                                .EQUIV  BIT02,BIT2
                                .EQUIV  BIT01,BIT1
                                .EQUIV  BIT00,BIT0

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

                                .SBTTL  TRAP CATCHER

000000                                  .=0
                                ;*ALL UNUSED LOCATIONS FROM 4 - 776 CONTAIN A ".+2,HALT"
                                ;*SEQUENCE TO CATCH ILLEGAL TRAPS AND INTERRUPTS
                                ;*LOCATION 0 CONTAINS 0 TO CATCH IMPROPERLY LOADED VECTORS
000174                                  .=174
000174  000000                 DISPREG: .WORD  0               ;;SOFTWARE DISPLAY REGISTER
000176  000000                 SWREG:   .WORD  0               ;;SOFTWARE SWITCH REGISTER
                                .SBTTL  STARTING ADDRESS(ES)
000200  000137  004216                  JMP     @#BEGIN ;;JUMP TO STARTING ADDRESS OF PROGRAM
000210                                  .=210
000210  000137  004206                  JMP     @#BEGIN2        ;JUMP SELECT TEST
                                ;*STARTING ADDRESS 200 FOR NORMAL STARTS
                                ;*THIS WILL TEST ALL RP04'S ON THE SYSTEM A SINGLE DRIVE AT A TIME
                                ;*
                                ;*STARTING ADDRESS 210 WILL TEST ONLY ONE SPECIFIED DRIVE
                                .SBTTL  MEMORY MANAGEMENT DEFINITIONS

                                ;*KT11 VECTOR ADDRESS

000250                          MMVEC=  250

                                ;*KT11 STATUS REGISTER ADDRESSES

177572                          SR0=    177572
177574                          SR1=    177574
177576                          SR2=    177576
172516                          SR3=    172516

                                ;*KERNEL "I" PAGE DESCRIPTOR REGISTERS
```

```
                        172300          KIPDR0= 172300
                        172302          KIPDR1= 172302
                        172304          KIPDR2= 172304
                        172306          KIPDR3= 172306
                        172310          KIPDR4= 172310
                        172312          KIPDR5= 172312
                        172314          KIPDR6= 172314
                        172316          KIPDR7= 172316

                                        ;*KERNEL "I" PAGE ADDRESS REGISTERS

                        172340          KIPAR0= 172340
                        172342          KIPAR1= 172342
                        172344          KIPAR2= 172344
                        172346          KIPAR3= 172346
                        172350          KIPAR4= 172350
                        172352          KIPAR5= 172352
                        172354          KIPAR6= 172354
                        172356          KIPAR7= 172356

                                        ;;*****************************************************************
                        000046                  .=46
000046  036652                          $ENDAD
                        000052          .=52
000052  040000                          40000
                        001110          .=1110
```

```
                                .SBTTL  COMMON TAGS

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

                        001100          .=1100
001100                  $CMTAG:                         ;;START OF COMMON TAGS
001100  000000          $PASS:  .WORD   0               ;;CONTAINS PASS COUNT
001102  000             $TSTNM: .BYTE   0               ;;CONTAINS THE TEST NUMBER
001103  000             $ERFLG: .BYTE   0               ;;CONTAINS ERROR FLAG
001104  000000          $ICNT:  .WORD   0               ;;CONTAINS SUBTEST ITERATION COUNT
001106  000000          $LPADR: .WORD   0               ;;CONTAINS SCOPE LOOP ADDRESS
001110  000000          $LPERR: .WORD   0               ;;CONTAINS SCOPE RETURN FOR ERRORS
001112  000000          $ERTTL: .WORD   0               ;;CONTAINS TOTAL ERRORS DETECTED
001114  000             $ITEMB: .BYTE   0               ;;CONTAINS ITEM CONTROL BYTE
001115  001             $ERMAX: .BYTE   1               ;;CONTAINS MAX. ERRORS PER TEST
001116  000000          $ERRPC: .WORD   0               ;;CONTAINS PC OF LAST ERROR INSTRUCTION
001120  000000          $GDADR: .WORD   0               ;;CONTAINS ADDRESS OF 'GOOD' DATA
001122  000000          $BDADR: .WORD   0               ;;CONTAINS ADDRESS OF 'BAD' DATA
001124  000000          $GDDAT: .WORD   0               ;;CONTAINS 'GOOD' DATA
001126  000000          $BDDAT: .WORD   0               ;;CONTAINS 'BAD' DATA
001130  000000                  .WORD   0               ;;RESERVED--NOT TO BE USED
001132  000000                  .WORD   0
001134  000             $AUTOB: .BYTE   0               ;;AUTOMATIC MODE INDICATOR
001135  000             $INTAG: .BYTE   0               ;;INTERRUPT MODE INDICATOR
001136  000000                  .WORD   0
001140  177570          SWR:    .WORD   DSWR            ;;ADDRESS OF SWITCH REGISTER
001142  177570          DISPLAY: .WORD  DDISP           ;;ADDRESS OF DISPLAY REGISTER
001144  177560          $TKS:   177560                  ;;TTY KBD STATUS
001146  177562          $TKB:   177562                  ;;TTY KBD BUFFER
001150  177564          $TPS:   177564                  ;;TTY PRINTER STATUS REG. ADDRESS
001152  177566          $TPB:   177566                  ;;TTY PRINTER BUFFER REG. ADDRESS
001154  000             $NULL:  .BYTE   0               ;;CONTAINS NULL CHARACTER FOR FILLS
001155  002             $FILLS: .BYTE   2               ;;CONTAINS # OF FILLER CHARACTERS REQUIRED
001156  012             $FILLC: .BYTE   12              ;;INSERT FILL CHARS. AFTER A "LINE FEED"
001157  000             $TPFLG: .BYTE   0               ;;"TERMINAL AVAILABLE" FLAG (BIT<07>=0=YES)
001160  000000          $REGAD: .WORD   0               ;;CONTAINS THE ADDRESS FROM
                                                        ;;WHICH  ($REG0) WAS OBTAINED
001162  000000          $REG0:  .WORD   0               ;;CONTAINS (($REGAD)+0)
001164  000000          $REG1:  .WORD   0               ;;CONTAINS (($REGAD)+2)
001166  000000          $REG2:  .WORD   0               ;;CONTAINS (($REGAD)+4)
001170  000000          $REG3:  .WORD   0               ;;CONTAINS (($REGAD)+6)
001172  000000          $REG4:  .WORD   0               ;;CONTAINS (($REGAD)+10)
001174  000000          $REG5:  .WORD   0               ;;CONTAINS (($REGAD)+12)
001176  000000          $TMP0:  .WORD   0               ;;USER DEFINED
001200  000000          $TMP1:  .WORD   0               ;;USER DEFINED
001202  000000          $TMP2:  .WORD   0               ;;USER DEFINED
001204  000000          $TMP3:  .WORD   0               ;;USER DEFINED
001206  000000          $TMP4:  .WORD   0               ;;USER DEFINED
001210  000000          $TMP5:  .WORD   0               ;;USER DEFINED
001212  000000          $TIMES: 0                       ;;MAX. NUMBER OF ITERATIONS
001214  000000          $ESCAPE:0                       ;;ESCAPE ON ERROR ADDRESS
001216  177607  000377    $BELL:  .ASCIZ  <207><377><377> ;;CODE FOR BELL
001222  077             $QUES:  .ASCII  /?/             ;;QUESTION MARK
001223  015             $CRLF:  .ASCII  <15>            ;;CARRIAGE RETURN
```

```
 2007  001224  000012                  $LF:    .ASCIZ  <12>            ;;LINE FEED
 2008                                  ;;*****************************************************************
```

```
                                .SBTTL  ERROR POINTER TABLE

                                ;*THIS TABLE CONTAINS THE INFORMATION FOR EACH ERROR THAT CAN OCCUR.
                                ;*THE INFORMATION IS OBTAINED BY USING THE INDEX NUMBER FOUND IN
                                ;*LOCATION $ITEMB. THIS NUMBER INDICATES WHICH ITEM IN THE TABLE IS PERTINENT.
                                ;*NOTE1:        IF $ITEMB IS 0 THE ONLY PERTINENT DATA IS ($ERRPC).
                                ;*NOTE2:       EACH ITEM IN THE TABLE CONTAINS 4 POINTERS EXPLAINED AS FOLLOWS:

                                ;*      EM              ;;POINTS TO THE ERROR MESSAGE
                                ;*      DH              ;;POINTS TO THE DATA HEADER
                                ;*      DT              ;;POINTS TO THE DATA
                                ;*      DF              ;;POINTS TO THE DATA FORMAT

001226                          $ERRTB:

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

                                ;ITEM1
001226  053706                          EM1                     ;WRONG DATA IN READING OR WRITING HARDWARE REGISTER
001230  056477                          DH1                     ;PC
                                                                ;REG. ADDR.
                                                                ;GOOD DATA
                                                                ;RECEIVED DATA
001232  062620                          DT1                     ;$ERRPC,REGADR,$GDDAT,$BDDAT
001234  063320                          DF1                     ;0,0,0,0,0

                                ;ITEM2
001236  053771                          EM2                     ;ERROR ON  DATA COMMAND

001240  061463                          DH33                    ;PC
                                                                ;PC OF JSR
                                                                ;TEST NO
                                                                ;WORD NO.
                                                                ;GOOD DATA
                                                                ;CONTENTS OF RHCS1
                                                                ;CONTENTS OF RHDS1
                                                                ;CONTENTS OF RHER1
001242  063172                          DT33                    ;$ERRPC,PCJSR,$TSTNM,ERWORD,$GDDAT,CS1,DS1,ER1
001244  063463                          DF33                    ;0,0,0,1,0,0,0,0

                                ;ITEM3
001246  053771                          EM2                     ;ERROR ON  DATA COMMAND

001250  061240                          DH32                    ;PC
                                                                ;PC OF JSR
                                                                ;TEST NO
                                                                ;WORD NO.
                                                                ;GOOD DATA
```

```
                                                                ;BAD DATA
                                                                ;CONTENTS OF RHCS1
                                                                ;CONTENTS OF RHDS1
                                                                ;CONTENTS OF RHER1

001252  063146                          DT32                    ;$ERRPC,PCJSR,$TSTNM,ERWORD,$GDDAT,$BDDAT,CS1,DS1,ER1
001254  063452                          DF32                    ;0,0,0,1,0,0,0,0,0,

                                ;ITEM4
001256  053771                          EM2                     ;ERROR ON  DATA COMMAND

001260  061036                          DH31                    ;PC
                                                                ;TEST NO
                                                                ;WORD NO.
                                                                ;GOOD DATA
                                                                ;BAD DATA
                                                                ;CONTENTS OF RHCS1
                                                                ;CONTENTS OF RHDS1
                                                                ;CONTENTS OF RHER1

001262  063124                          DT31                    ;$ERRPC,$TSTNM,ERWORD,$GDDAT,$BDDAT,CS1,DS1,ER1
001264  063442                          DF31                    ;0,0,1,0,0,0,0,0,

                                ;ITEM5
001266  000000                          0                       ;
001270  000000                          0                       ;
001272  063124                          DT31                    ;$ERRPC,$TSTNM,ERWORD,$GDDAT,$BDDAT,CS1,DS1,ER1
001274  063442                          DF31                    ;0,0,1,0,0,0,0,0,

                                ;ITEM6
001276  054020                          EM6                     ;ERROR ON WRITE HEADER AND DATA

001300  061240                          DH32                    ;PC
                                                                ;PC OF JSR
                                                                ;TEST NO
                                                                ;WORD NO.
                                                                ;GOOD DATA
                                                                ;BAD DATA
                                                                ;CONTENTS OF RHCS1
                                                                ;CONTENTS OF RHDS1
                                                                ;CONTENTS OF RHER1

001302  063146                          DT32                    ;$ERRPC,PCJSR,$TSTNM,ERWORD,$GDDAT,$BDDAT,CS1,DS1,ER1
001304  063452                          DF32                    ;0,0,0,1,0,0,0,0,0,

                                ;ITEM7
001306  054020                          EM6                     ;ERROR ON WRITE HEADER AND DATA
001310  056622                          DH2                     ;PC
                                                                ;TEST NO
                                                                ;WORD NO.
```

```
                                                                ;GOOD DATA
                                                                ;BAD DATA
001312  062646                          DT3                     ;$ERRPC,$TSTNM,ERWORD,$GDDAT,$BDDAT
001314  063331                          DF3                     ;0,0,1,0,0,

                                ;ITEM10
001316  000000                          0                       ;
001320  000000                          0                       ;
001322  062646                          DT3                     ;$ERRPC,$TSTNM,ERWORD,$GDDAT,$BDDAT
001324  063331                          DF3                     ;0,0,1,0,0,

                                ;ITEM11
001326  054057                          EM11                    ;CONTROLLER OR DRIVE STATUS
001330  056745                          DH11                    ;PC
                                                                ;TEST NO
                                                                ;FAILING REG. ADDR
                                                                ;CONTENTS OF RHCS1
                                                                ;CONTENTS OF RHCS2
                                                                ;CONTENTS OF RHDS1
                                                                ;CONTENTS OF RHER1
001332  062662                          DT11                    ;$ERRPC,$TSTNM,$BDADR,CS1,CS2,DS1,ER1
001334  063336                          DF11                    ;0,0,0,0,0,0

                                ;ITEM12
001336  054057                          EM11                    ;WRONG DATA FROM SILO

001340  056477                          DH1                     ;PC
                                                                ;REG.ADDR
                                                                ;GOOD DATA
                                                                ;RECEIVED DATA
001342  062620                          DT1                     ;$ERRPC,REGADR,$GDDAT,$BDDAT
001344  063320                          DF1                     ;0,0,0,0

                                ;ITEM13
001346  000000                          0
001350  000000                          0
001352  062620                          DT1                     ;$ERRPC,TSTNM,REGADR,$GDDAT,$BDDAT
001354  063320                          DF1                     ;0,0,0,0,0

                                ;ITEM14
001356  054112                          EM14                    ;REGISTER FAILED
001360  057124                          DH14                    ;PC
                                                                ;FAILING REG. ADDR
                                                                ;CONTENTS OF FAILING REG.
                                                                ;CONTENTS OF RHCS1
                                                                ;CONTENTS OF RHCS2
                                                                ;CONTENTS OF RHDS1
                                                                ;CONTENTS OF RHER1
001362  062702                          DT14                    ;$ERRPC,$BDADR,$BDDAT,CS1,CS2,DS1,ER1
001364  063345                          DF14                    ;0,0,0,0,0,0,0
```

```
                                ;ITEM15
001366  054132                          EM15            ;SPECIFIED REG. NON EXISTANT SO ABORT
                                                        ;PROGRAM
001370  057326                          DH15            ;PC
                                                        ;ADDR. OF REG
001372  062724                          DT15            ;$ERRPC,TEMP1
001374  063355                          DF15            ;0,0


                                ;ITEM16
001376  054203                          EM16            ;WAIT LOOP FAILED
001400  057350                          DH16            ;PC
                                                        ;WAT PC
                                                        ;BIT WANTED
                                                        ;REG. ADR.
                                                        ;REG. CONT.
001402  062732                          DT16            ;$ERRPC,$TMP3,$TMP1,$TMP0,$BDDAT
001404  063357                          DF16            ;0,0,0,0


                                ;ITEM17
001406  054224                          EM17            ;WRITE CHECK FAILING
001410  057467                          DH17            ;PC
                                                        ;TEST NO
                                                        ;CONTENTS OF RHBA
                                                        ;CONTENTS OF RHDB
                                                        ;CONTENTS OF RHWC
                                                        ;CONTENTS OF RHCS1
                                                        ;CONTENTS OF RHCS2
001412  062746                          DT17            ;$ERRPC,$TSTNM,$BA,DB,WC,CS1,CS2
001414  063363                          DF17            ;0,0,0,0,0,0,0


                                ;ITEM20
001416  054250                          EM20            ;REGISTER FAILING
001420  057652                          DH20            ;PC
                                                        ;TST NO
                                                        ;CONTENTS OF RHER1
                                                        ;CONTENTS OF RHER2
                                                        ;CONTENTS OF RHER3
                                                        ;CONTENTS OF RHAS
                                                        ;CONTENTS OF RHDS1
001422  062766                          DT20            ;$ERRPC,TSTNM ER1,ER2,ER3,AS,DS1
001424  063372                          DF20            ;0,0,0,0,0,0,0


                                ;ITEM21
001426  054271                          EM21            ;INTERRUPT FAILING
001430  060026                          DH21            ;PC
                                                        ;TEST NO
                                                        ;CONTENTS OF RHCS1
                                                        ;CONTENTS OF RHAS
                                                        ;CONTENTS OF RHDS1
001432  063006                          DT21            ;$ERRPC,TSTNM,CS1,AS,DS1
001434  063401                          DF21            ;0,0,0,0,0
```

```
                                ;ITEM22
001436  054313                          EM22            ;MISSMATCH IN DRIVE PRESENT
                                                        ;LOOKING AT RHAS AND RHCS2-NED(BIT#12)
                                                        ;DRIVE PRESENT DO NOT AGREE
                                                        ;NOTE: ON DUAL PORT SYSTEM
                                                        ;DRIVE ON OTHER PORT WILL NOT GIVE NED
                                                        ;HENCE THERE WILL BE A MISSMATCH
                                                        ;177777-MEANS NOT PRESENT
001440  060147                          DH22            ;PC
                                                        ;TEST NO
                                                        ;RHAS UNIT
                                                        ;RHCS2 UNIT
                                                        ;
001442  063022                          DT22            ;$ERRPC,TSTNM$,$GDDAT,$BDDAT
001444  063406                          DF22            ;0,0,0,0

                                ;ITEM23
001446  000000                          0               ;MISSMATCH IN DRIVE PRESENT
                                                        ;LOOKING AT RHAS AND RHCS2-NED(BIT#12)
                                                        ;DRIVE PRESENT DO NOT AGREE
                                                        ;177777-MEANS NOT PRESENT
001450  000000                          0               ;PC
                                                        ;TEST NO
                                                        ;RHAS UNIT
                                                        ;RHCS2 UNIT
                                                        ;
001452  063022                          DT22            ;$ERRPC,TSTNM$,$GDDAT,$BDDAT
001454  063406                          DF22            ;0,0,0,0


                                ;ITEM 24
001456  054706                          EM24            ;LOOK AHEAD REGISTER AT THE
                                                        ;BEGINNING OF A SECTOR IS IN
                                                        ;ERROR
001460  060246                          DH24            ;PC
                                                        ;RHDST
                                                        ;BAD RHLA
                                                        ;GOOD RHLA
                                                        ;SECTOR NO
                                                        ;SECTOR CLOCK
001462  063034                          DT24            ;$ERRPC,DST,$BDDAT,$TMP1,$TMP2,$TMP3
001464  063412                          DF24            ;0,0,0,0,0

                                ;ITEM 25
001466  055001                          EM25            ;LOOK AHEAD REGISTER IS
                                                        ;IN ERROR

001470  060246                          DH24            ;PC
                                                        ;RHDST
                                                        ;BAD RHLA
                                                        ;GOOD RHLA
                                                        ;SECTOR NO
```

```
                                                        ;SECTOR CLOCK
001472  063034                      DT24              ;$ERRPC,DST,$BDDAT,$TMP1,$TMP2,$TMP3
001474  063412                      DF24              ;0,0,0,0,0
                            ;ITEM26
001476  054057                      EM11              ;CONTROLLER OR DRIVE STATUS

001500  060404                      DH26              ;PC
                                                      ;PC OF JSR
                                                      ;FAILING REGISTER ADDRESS
                                                      ;CONTENTS OF RHCS1
                                                      ;CONTENTS OF RHCS2
                                                      ;CONTENTS OF RHDS1
                                                      ;CONTENTS OF RHER1

001502  063052                      DT26              ;$ERRPC,PCJSR,$BDADR,CS1,CS2,DS1,ER1
001504  063420                      DF26              ;0,0,0,0,0,0,

                            ;ITEM27
001506  053706                      EM1               ;ERROR IN READING OR WRITING HARDWARE REGISTER

001510  060566                      DH27              ;PC
                                                      ;PC OF JSR
                                                      ;TEST NUMBER
                                                      ;FAILING REGISTER
                                                      ;GOOD DATA
                                                      ;RECEIVED DATA

001512  063072                      DT27              ;$ERRPC,PCJSR,TSTNM,REGADR,$GDDAT,$BDDAT
001514  063427                      DF27              ;0,0,0,0,0,0

                            ;ITEM30
001516  055041                      EM30              ;CURRENT CYLINDER DOES NOT REFLECT DESIRED CYLINDER REG.
001520  060725                      DH30              ;PC
                                                      ;PC OF JSR
                                                      ;REGISTER ADDRESS
                                                      ;GOOD DATA
                                                      ;BAD DATA

001522  063110                      DT30              ;$ERRPC,PCJSR,REGADR,$GDDAT,$BDDAT
001524  063435                      DF30              ;0,0,0,0,0

                            ;ITEM31
001526  055163                      EM31              ;ECC GENERATED IS INCORRECT
                                                      ;EVERY WORD IN THIS SECTOR IS GIVEN IN "DATA USED"

001530  061674                      DH34              ;PC
                                                      ;TEST NUMBER
                                                      ;GOOD ECC1
                                                      ;GOOD EC2C
                                                      ;WRITTEN ECC1
```

```
                                                                ;WRITTEN ECC2
                                                                ;DATA USED
                                                        
001532  063214                          DT34                    ;$ERRPC,TSTNM,GECC1,GECC2,WECC1,WECC2,DISK
                                                        
001534  063473                          DF34                    ;0,0,0,0,0,0,0
                                                        
                                                        
                                ;ITEM32
001536  055306                          EM32                    ;ON READ COMMAND AFTER DATA AND ECC HAVE BEEN READ
                                                                ;ECC REGISTER OR RHER1 IS IN ERROR
                                                                ;ONLY LOWER 11 BITS OF PATTERN REGISTER
                                                                ;CAN BE READ
                                                                ;THIS SHUOLD MATCH LOWER 11 BITS OF ECC1
                                                        
001540  062047                          DH35                    ;PC
                                                                ;TEST NUMBER
                                                                ;GOOD ECC1
                                                                ;GOOD ECC2
                                                                ;PATTERN REGISTER
                                                                ;RHER1
                                                        
001542  063234                          DT35                    ;$ERRPC,TSTNM,GECC1,GECC2,EC2,ER1
                                                        
001544  063502                          DF35                    ;0,0,0,0,0,0
                                                        
                                ;ITEM33
001546  055572                          EM33                    ;HIGH COUNT BIT NOT HIGH AFTER 38859 CLOCKS
001550  062243                          DH36                    ;PC
                                                                ;PC OF JSR
                                                                ;TEST NUMBER
                                                                ;RHMR
                                                                ;POSITION REG.
                                                                ;PATTERN REGISTER
                                                        
001552  063256                          DT36                    ;$ERRPC,PCJSR,TSTNM,MR,EC1,EC2
                                                        
001554  063512                          DF36                    ;0,0,0,0,0,0
                                                        
                                ;ITEM34
001556  055644                          EM34                    ;ZERO DETECT BIT NOT HIGH WHEN THE
                                                                ;32 BIT ECC REGISTER HAS ITS 21 BITS
                                                                ;OF ZEROS
                                                                ;ERROR PRINTOUT WILL CONTINUE TILL
                                                                ;ZERO DETECT BIT IS HIGH
001560  062243                          DH36                    ;PC
                                                                ;PC OF JSR
                                                                ;TEST NUMBER
                                                                ;RHMR
                                                                ;POSITION REG.
                                                                ;PATTERN REGISTER
                                                        
001562  063256                          DT36                    ;$ERRPC,PCJSR,TSTNM,MR,EC1,EC2
```

```
2401  001564  063512                        DF36            ;0,0,0,0,0,0
2402
2403
2404
2405                                ;ITEM35
2406  001566  055737                        EM35            ;POSITION REGISTER OR 11 BITS OF
2407                                                          ;PATTERN REGISTER INCORRECT
2408                                                          ;LOWER 11 BITS OF PATTERN REGISTER
2409                                                          ;SHOULD MATCH LOWER 11 BITS OF GOOD ECC1
2410                                                          ;DATA ENVELOPE AND N-CODE ZEROS ARE IN DECIMAL
2411
2412  001570  062401                        DH37            ;PC
2413                                                          ;TEST NUMBER
2414                                                          ;ECC POSITION
2415                                                          ;GOOD POSITION
2416                                                          ;GOOD ECC1
2417                                                          ;GOOD ECC2
2418                                                          ;ECC PATTERN
2419                                                          ;DATA ENVELOPE
2420                                                          ;N-CODE ZEROS
2421
2422  001572  063274                        DT37            ;$ERRPC,TSTNM,EC1,POSITI,GECC1,GECC2,EC2,DATENV,ZCODE
2423
2424  001574  063520                        DF37            ;0,0,0,0,0,0,0,0,0
2425
2426
2427
2428                                ;ITEM36
2429  001576  056236                        EM36            ;ON A READ COMMAND WITH NON CORRECTABLE
2430                                                          ;ERROR INSERTED DCK AND ECH SHOULD BE SET
2431  001600  062047                        DH35            ;PC
2432                                                          ;TEST NUMBER
2433                                                          ;GOOD ECC1
2434                                                          ;GOOD ECC2
2435                                                          ;PATTERN REGISTER
2436                                                          ;POSITION REGISTER
2437                                                          ;RHER1
2438
2439  001602  063234                        DT35            ;$ERRPC,TSTNM,GECC1,GECC2,EC2,EC1,ER1
2440
2441  001604  063502                        DF35            ;0,0,0,0,0,0,0
2442
2443                                ;ITEM37
2444  001606  056344                        EM37        ;PGE ERROR
2445  001610  056745                        DH11                ;PC
2446                                                          ;TEST NO
2447                                                          ;FAILING REG. ADDR
2448                                                          ;CONTENTS OF RHCS1
2449                                                          ;CONTENTS OF RHCS2
2450                                                          ;CONTENTS OF RHDS1
2451                                                          ;CONTENTS OF RHER1
2452  001612  062662                        DT11            ;$ERRPC,$TSTNM,$BDADR,CS1,CS2,DS1,ER1
2453  001614  063336                        DF11            ;0,0,0,0,0,0
2454
2455
2456
```

```
2457
2458
2459
2460
2461
```

```
                        ;;*****************************************************************
                        ;RH11 REGISTERS



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


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


                        ;CONTROL AND STATUS REGISTER 2 (RHCS2)

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

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


                        ;;*****************************************************************
                        ;RP04 REGISTERS
                        ;;*****************************************************************


                        ;CONTROL AND STATUS 1 REGISTER. (#00)

000001                  GO=     1               ;GO (BIT #0)
000100                  IE=     100             ;INTERRUPT ENABLE (BIT #6)
000200                  RDY=    200             ;READY (BIT #7)
000400                  A16=    400             ;HIGH ORDER UNIBUS BITS (BIT #8)
001000                  A17=    1000            ;HIGH ORDER UNIBUS BITS (BIT #9)
000000                  PSEL=   0               ;PORT SELECT (BIT #10)
004000                  DVA=    4000            ;DEVICE AVAILABLE (BIT #11)
020000                  MCPE=   20000           ;MASSBUSS PARITY ERROR (BIT #13)
```

```
                        TRE=    40000           ;TRANSFER ERROR (BIT #14)
                        SC=     100000          ;SPECIAL CONDITION (BIT #15)

                        ;STATUS REGISTER (RHDS1) (#01)

000001                  DFS=    1               ;DRIVE FORWARD 5"/SEC.  (BIT #0)
000002                  DFF20=  2               ;DRIVE FORWARD 20"/SEC.  (BIT #1)
000004                  DIGB=   4               ;DRIVE TO INNER GAVRD BAND (BIT #2)
000010                  GRV=    10              ;GO REVERSE (BIT #3)
000020                  DL64=   20              ;DIFFERENCE LESS THAN 64 (BIT #4)
000040                  DE1=    40              ;DIFFERENCE EQUALS 1 (BIT #5)
000100                  VV=     100             ;VOLUME VALID (BIT #6)
000200                  DRY=    200             ;DRIVE READY (BIT #7)
000400                  DPR=    400             ;DRIVE PRESENT (BIT #8)
001000                  PROG=   1000            ;PROGRAMABLE (BIT #9)
002000                  LST=.   2000            ;LAST SECTOR TRANSFERRED (BIT #10)
004000                  WRL=    4000            ;WRITE LOCK (BIT #11)
010000                  MOL=    10000           ;MEDIUM ON-LINE (BIT #12)
020000                  PIP=    20000           ;POSITIONING OPERATION IN PROGRESS (BIT #13)
040000                  ERR=    40000           ;COMPOSIT ERROR. (BIT #14)
100000                  ATA=    100000          ;ATTENTION ACTIVE (BIT #15)

                        ;ERROR REGISTER #01 (RHER1) (#02)
000001                  ILF=    1               ;ILLEGAL FUNCTION (BIT #0)
000002                  ILR=    2               ;ILLEGAL REGISTER (BIT #1)
000004                  RMR=    4               ;REGISTER MODIFICATION REFUSED (BIT #2)
000010                  PAR=    10              ;PARITY ERROR (BIT #3)
000020                  FER=    20              ;FORMAT ERROR (BIT #4)
000040                  WCF=    40              ;WRITE CLOCK FAIL (BIT #5)
000100                  ECH=    100             ;ECC HARD ERROR (BIT #6)
000200                  HCE=    200             ;HEADER COMPARE ERROR (BIT #7)
000400                  HCRC=   400             ;HEADER CRC ERROR (BIT #8)
001000                  AOE=    1000            ;ADDRESS OVERFLOW ERROR (BIT #9)
002000                  IAE=    2000            ;INVALID ADDRESS ERROR (BIT #10)
004000                  WLE=    4000            ;WRITE LOCK ERROR (BIT #11)
010000                  DTE=    10000           ;DRIVE TIMING ERROR (BIT #12)
020000                  OPI=    20000           ;OPERATION INCOMPLETE (BIT #13)
040000                  UNS=    40000           ;DRIVE UNSAFE (BIT #14)
100000                  DCK=    100000          ;DATA CHECK ERROR (BIT 15)

                        ;MAINTAINABILITY REGISTER (RHMR)(#03)

000001                  DMD=    1               ;DIAGINOSTIC MODE (BIT #0)
000002                  MCLK=   2               ;MAINTAINABILITY CLOCK (BIT #1)
000004                  MINX=   4               ;MAINTAINABILITY INDEX (BIT #2)
000010                  MSTCK=  10              ;MAINTAINABILITY SECTOR CLOCK (BIT #3)
000020                  MRD=    20              ;MAINTAINABILITY READ (BIT #4)
000040                  MWR=    40              ;MAINTAINABILITY WRITE (BIT #5)
000200                  DENVL=  200             ;DATA ENVELOPE (BIT #7)
000400                  ZER=    400             ;ZERO DETECT (BIT #8)
001000                  DTSY=   1000            ;MAINTAINABILITY SYNC DETECTED (BIT #9)

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

000001                  AT0=    1               ;DEVICE 0 (BIT #0)
000002                  AT1=    2               ;DEVICE 1 (BIT #1)
```

```
2574            000004                  AT2=    4                       ;DEVICE 2 (BIT #2)
2575            000010                  AT3=    10                      ;DEVICE 3 (BIT #3)
2576            000020                  AT4=    20                      ;DEVICE 4 (BIT #4)
2577            000040                  AT5=    40                      ;DEVICE 5 (BIT #5)
2578            000100                  AT6=    100                     ;DEVICE 6 (BIT #6)
2579            000200                  AT7=    200                     ;DEVICE 7 (BIT #7)
2580
2581                                    ;DESIRED SECTOR/TRACK ADDRESS REGISTER (RHDST) (#1)
2582                                    ;EACH BIT IS CALLED BY BIT NUMBER
2583                                    ;DRIVE TYPE REGISTER (RHDT) (#06)
2584                                    ;EACH BIT IS CALLED BY BIT NUMBER
2585                                    ;LOOK-AHEAD REGISTER (RHLA) (#07)
2586
2587            000001                  EXT1=   1                       ;EXTENSION 1 (BIT #0)
2588            000002                  EXT2=   2                       ;EXTENSION 2 (BIT #1)
2589            000004                  EXT4=   4                       ;EXTENSION 3 (BIT #2)
2590            000010                  EXT10=  10                      ;EXTENSION 4 (BIT #3)
2591            000020                  EXT20=  20                      ;EXTENSION 5 (BIT #4)
2592            000040                  EXT40=  40                      ;EXTENSION 6 (BIT #5)
2593            000100                  SC1=    100                     ;SECTOR COUNT FIELD 0 (BIT #6)
2594            000200                  SC2=    200                     ;SECTOR COUNT FIELD 1 (BIT #7)
2595            000400                  SC4=    400                     ;SECTOR COUNT FIELD 2 (BIT #8)
2596            001000                  SC10=   1000                    ;SECTOR COUNT FIELD 3 (BIT #9)
2597            002000                  SC20=   2000                    ;SECTOR COUNT FIELD 4 (BIT #10)
2598            004000                  TRK1=   4000                    ;TRACK FIELD 1 (BIT #11)
2599            010000                  TRK2=   10000                   ;TRACK FIELD 2 (BIT #12)
2600            020000                  TRK4=   20000                   ;TRACK FIELD 3 (BIT #13)
2601            040000                  TRK10=  40000                   ;TRACK FIELD 4 (BIT #14)
2602            100000                  TRK20=  100000                  ;TRACK FIELD 5 (BIT #15)
2603
2604                                    ;ERROR REGISTER #2 (RHER2) (#10)
2605
2606            000001                  WCU=    1                       ;WRITE CURRENT UNSAFE (BIT #0)
2607            000002                  CSF=    2                       ;CURRENT SINK FAILURE (BIT #1)
2608            000004                  WSU=    4                       ;WRITE SELECT UNSAFE (BIT #2)
2609            000010                  CSU=    10                      ;CURRENT SWITCH UNSAFE (BIT #3)
2610            000020                  MSE=    20                      ;MOTOR SEQUENCE ERROR (BIT #4)
2611            000040                  TDF=    40                      ;TRANSITIONS DETECTOR FAILURE (BIT #5)
2612            000100                  TUF=    100                     ;TRANSITIONS UNSAFE (BIT #6)
2613            000200                  FEN=    200                     ;FAILSAFE ENABLED (BIT #7)
2614            000400                  WRU=    400                     ;WRITE READY UNSAFE (BIT #8)
2615            001000                  MHS=    1000                    ;MULTIPLE HEAD SELECT (BIT #9)
2616            002000                  NHS=    2000                    ;NO HEAD SELECTION (BIT #10)
2617            004000                  IXE=    4000                    ;INDEX ERROR (BIT #11)
2618            010000                  VU30=   10000                   ;30VOLT UNSAFE (BIT #12)
2619            020000                  PLU=    20000                   ;PLO UNSAFE (BIT #13)
2620            100000                  ACU=    100000                  ;ACUNSAFE (BIT #15)
2621
2622                                    ;OFFSET REGISTER (RHOF) (#11)
2623
2624            000001                  OF25=   1                       ;OFFSET 25 MICRO INCHES (BIT #0)
2625            000002                  OF50=   2                       ;OFFSET 50 MICRO INCHES (BIT #1)
2626            000004                  OF100=  4                       ;OFFSET 100 MICRO INCHES (BIT #2)
2627            000010                  OF200=  10                      ;OFFSET 200 MICRO INCHES (BIT #3)
2628            000020                  OF400=  20                      ;OFFSET 400 MICRO INCHES (BIT #4)
2629            000040                  OF800=  40                      ;OFFSET 800 MICRO INCHES (BIT #5)
```

```
2630
2631            000200          OFREV=  200             ;OFFSET NEGATIVE (REVERSE) (BIT #7)
2632            002000          HCI=    2000            ;HEADER COMPARE INHIBIT (BIT #10)
2633            004000          ECI=    4000            ;ERROR CORRECTION CODE INHIBIT (BIT #11)
2634            010000          FMT22=  10000           ;FORMAT BIT (BIT #12
2635
2636                            ;DESIRED CYLINDER ADDRESS (RHCA) (#12)
2637                            ;EACH BIT IS CALLED BY BIT NUMBER.
2638                            ;CURRENT CYLINDER ADDRESS (RHCC) (#13)
2639                            ;EACH BIT IS CALLED BY BIT NUMBER
2640                            ;SERIAL NUMBER REGISTER (RHSN) (#14)
2641                            ;EACH IS CALLED BY BIT NUMBER
2642                            ;ERROR REGISTER #03 (RHER3) (#15)
2643
2644            000001          PSU=    1               ;PACK SPEED UNSAFE (BIT #0)
2645            000002          VUF=    2               ;VELOCITY UNSAFE (BIT #1)
2646            000010          UWR=    10              ;ANY UNSAFE EXCEPT READ/WRITE (BIT #3)
2647            000020          PRE=    20              ;DISK PACK ROTATION ERROR (BIT #4)
2648            000040          ACL=    40              ;AC LOW (BIT #5)
2649            000100          DCL=    100             ;DC LOW (BIT #6)
2650            040000          SKI=    40000           ;SEEK INCOMPLETE (BIT #14)
2651            100000          OCYL=   100000          ;OFF CYLINDER (BIT #15)
2652
2653                            ;ECC POSITION REGISTER (RHEC1) (#16)
2654                            ;EACH BIT IS CALLED BY BIT NUMBER
2655                            ;ECC PATTERN REGISTER (RHEC2) (#17)
2656                            ;EACH BIT IS CALLED BY BIT NUMBER
```

```
                                .SBTTL  REGISTER ADDRESSES

                                ;RP04 VECTOR ADDRESS
001616  000254                  RPVEC:  254                     ;RP04 VECTOR ADDRESS

                                ;RP04 DISK I/O REGISTERS LOCATED IN THE RH11 CONTROLLER
                                ;NOTE:  THE CONTENTS OF THESE LOCATIONS WILL BE DIFFRENT
                                ;       IF THE "CHANGE BASE ADDRESS" ROUTINE IS USED.
                                ;       THIS ROUTINE STARTS AT LOCATION TAGED "BASECH"
001620  176722                  RHDB:   176722                  ;DATA BUFFER SEE NOTE ABOVE
001622  176702                  RHWC:   176702                  ;WORD COUNT SEE NOTE ABOVE
001624  176704                  RHBA:   176704                  ;BUS ADDRESS SEE NOTE ABOVE
001626  176710                  RHCS2:  176710                  ;CONTROL AND STATUS 2 SEE NOTE ABOVE

                                ;RP04 DISK I/O REGISTERS LOCATED IN THE RP04 DEVICE LOGIC
                                ;NOTE:  THE CONTENTS OF THESE LOCATIONS WILL BE DIFFRENT
                                ;       IF THE "CHANGE BASE ADDRESS ROUTINE IS USED.
                                ;       THIS ROUTINE STARTS AT LOCATION TAGED "BASECH"
001630  176700                  RHCS1:  176700                  ;CONTROL AND STATUS 1 SEE NOTE ABOVE
001632  176714                  RHER1:  176714                  ;ERROR #1 SEE NOTE ABOVE
001634  176706                  RHDST:  176706                  ;DESIRED SECTOR/TRACK ADDRESS SEE NOTE ABOVE
001636  176740                  RHER2:  176740                  ;ERROR #2 SEE NOTE ABOVE
001640  176732                  RHOF:   176732                  ;OFFSET SEE NOTE ABOVE
001642  176734                  RHCA:   176734                  ;DESIRED CYLINDER ADDRESS SEE NOTE ABOVE
001644  176742                  RHER3:  176742                  ;ERROR #3 SEE NOTE ABOVE
001646  176716                  RHAS:   176716                  ;ATTENTION SUMMARY SEE NOTE ABOVE
001650  176724                  RHMR:   176724                  ;MAINTAINABILITY SEE NOTE ABOVE
001652  176712                  RHDS1:  176712                  ;DRIVE STATUS SEE NOTE ABOVE
001654  176726                  RHDT:   176726                  ;DRIVE TYPE SEE NOTE ABOVE
001656  176730                  RHSN:   176730                  ;SERIAL NUMBER SEE NOTE ABOVE
001660  176744                  RHEC1:  176744                  ;ECC POSITION SEE NOTE ABOVE
001662  176746                  RHEC2:  176746                  ;ECC PATTERN SEE NOTE ABOVE
001664  176720                  RHLA:   176720                  ;LOOK-AHEAD SEE NOTE ABOVE
001666  176736                  RHCC:   176736                  ;CURRENT CYLINDER ADDRESS SEE NOTE ABOVE
```

```
                                ;THE FOLLOWING LOCATIONS ARE RESERVED FOR REGISTERS
                                ;ANY TIME THERE IS AN ERROR ALL THESE WILL BE FILLED
                                ;ONLY SOME MAY BE PRINTED BUT ALL WILL BE FILLED TRUE
                                ;FOR THE TIME JUST AFTER THE "ERROR" ERROR COMMAND

001670  000000          DB:     0               ;DATA BUFFER
001672  000000          WC:     0               ;WORD COUNT
001674  000000          BA:     0               ;BUS ADDRESS
001676  000000          CS2:    0               ;CONTROL AND STATUS 2

001700  000000          CS1:    0               ;CONTROL AND STATUS 1
001702  000000          ER1:    0               ;ERROR #1
001704  000000          DST:    0               ;DESIRED SECTOR/TRACK ADDRESS
001706  000000          ER2:    0               ;ERROR #2
001710  000000          OF:     0               ;OFFSET
001712  000000          CA:     0               ;DESIRED CYLINDER ADDRESS
001714  000000          ER3:    0               ;ERROR #3
001716  000000          AS:     0               ;ATTENTION SUMMARY
001720  000000          MR:     0               ;MAINTAINABILITY
001722  000000          DS1:    0               ;DRIVE STATUS
001724  000000          DT:     0               ;DRIVE TYPE
001726  000000          SN:     0               ;SERIAL NUMBER
001730  000000          EC1:    0               ;ECC POSITION
001732  000000          EC2:    0               ;ECC PATTERN
001734  000000          LA:     0               ;LOOK-AHEAD
001736  000000          CC:     0               ;CURRENT CYLINDER ADDRESS

001740  000010          UNITS:  .BLKW   8.      ;THIS IS FILLED WITH -1
001760  000000          UNIT:   .WORD   0       ;UNIT UNDER TEST
001762  000000          NOUNIT: .WORD   0       ;NUMBER OF UNITS PRESENT
                                                ;USED TO KEEP TRACK OF UNIT UNDER TEST
001764  000000          NUNIT:  .WORD   0       ;USED TO DETERMIN IF THERE ARE MORE
                                                ;THAN ONE UNIT
001766  000000          SELECT: .WORD   0       ;ALL ONES INDICATE UNIT TO BE SELECTED
001770  000000          UNITSL: .WORD   0       ;UNIT NO. SELECTED

001772  000000          ERFLG$: 0               ;ERROR FLAG

001774  000000          SAVDT:  0               ;SAVE DRIVE TYPE REGISTER
                                                ;FOR COMPARISON IN DRIVE CLEAR TEST
                                                ;AND RH INIT TEST
001776  000000          SAVSN:  0               ;SAVE SERIAL NUMBER REGISTER
                                                ;FOR COMPARISON IN DRIVE CLEAR TEST
                                                ;AND RH INIT TEST

002000  000000          PCJSR:  0               ;SAVE PC OF JSR WHICH GAVE THE ERROR
```

```
2766
2767
2768
2769  002002  000000                ATTENT: 0           ;ATTENTION BIT FOR PRESENT UNIT
2770  002004  000000                TOTALAT:      0                 ;TATAL ATTENTION BITS
2771
2772
2773  002006  000000                TMPILL: 0                 ;TEMPORARY ILLEGAL FUNCTION
2774
2775
2776  002010  000000                TSECC:  0                 ;FLAG TO SAY IF ECC TEST OR NOT
2777                                                          ;WHEN =177777 IT IS AN ECC TEST
2778                                                          ;WHEN =0IT IS NOT AN ECC TEST
2779
2780  002012  000000                TESDTE: 0           ;FLAG TO SAY IF DRIVE TIMING ERROR OR NOT
2781                                                          ;WHEN = 177777 IT IS A DTE TEST
2782                                                          ;WHEN = 0      IT IS NOT A DTE TEST
2783
2784
2785  002014  000000                TAGDTE: 0                 ;TEMPORARY TAG USED IN DRIVE TIMING
2786                                                          ;ERROR TEST
```

```
                                ;FUNCTION EQUATES

                                        ;TABLE OF FUNCTIONS FOR RHCS1 THEN "GO" BIT HAS TO BE SET
002016                          FUTABL:
002016  000000                  NOPERA: 0                       ;NO OPERATION
002020  000002                  UNLOAD: 2                       ;UNLOAD (STAND BY)
002022  000006                  RECALI: 6                       ;RECALIBRATE
002024  000010                  DCLEAR: 10                      ;DRIVE CLEAR
002026  000012                  RELEAS: 12                      ;RELEASE (DUAL-PORT OPERATION)
002030  000030                  SERCH:  30                      ;SEARCH COMMAND
002032  000050                  WRCHEK: 50                      ;WRITE CHECK DATA
002034  000052                  WRCHDT: 52                      ;WRITE CHECK HEADER AND DATA
002036  000060                  WRIDAT: 60                      ;WRITE DATA
002040  000062                  WRIFOR: 62                      ;WRITE HEADER AND DATA (FORMAT)
002042  000070                  READAT: 70                      ;READ DATA
002044  000072                  REFOR:  72                      ;READ HEADER AND DATA
002046  000004                  SEECOM: 4                       ;SEEK COMMAND
002050  000014                  OFSETC: 14                      ;OFFSET COMMAND
002052  000016                  RETCL:  16                      ;RETURNTO CENTERLINE
002054  000022                  PKACK:  22                      ;PACK ACKNOWLEDGE
002056  000020                  READIN: 20                      ;READ IN
002060  000000                  ILLEGL: .WORD                   ;COMPUTED ILLEGAL FUNCTION

                                ;DATA BUFFER FOR READ WRITE
002062  000422                  WRFROM: .BLKW   274.            ;WRITE FROM THIS BUFFER
003126  000422                  REINTO: .BLKW   274.            ;READ INTO THIS BUFFER
004172  000000                  TSTNM:  0                       ;TEST NUMBER
004174  000000                  FIRST:  0                       ;IF ZERO WILL TYPE HEADER
                                                                ;IF ONES WILL NOT TYPE HEADER


                                ;TABLE FOR ATTENTION BITS
                                ;ATTENTION TABLE
004176  001     002     004     ATABLE: .BYTE   1,2,4,10,20,40,100,200
004201  010     020     040
004204  100     200
```

```
                                        .SBTTL  REGISTER TEST
004206  012737  177777  001766  BEGIN2: MOV     #-1,@#SELECT    ;SELECT UNIT
004214  000402                  BR      START
004216  005037  001766          BEGIN:  CLR     @#SELECT        ;DO NOT SELECT UNIT
                                                                ;NORMAL RUN

004222                          START:
                                .SBTTL  INITIALIZE THE COMMON TAGS
                                ;;CLEAR THE COMMON TAGS ($CMTAG) AREA
004222  012706  001100                  MOV     #$CMTAG,R6      ;;FIRST LOCATION TO BE CLEARED
004226  005026                  1$:     CLR     (R6)+           ;;CLEAR MEMORY LOCATION
004230  022706  001140                  CMP     #SWR,R6 ;;DONE?
004234  001374                  BNE     .-6                     ;;LOOP BACK IF NO
004236  012706  001000                  MOV     #STACK,SP       ;;SETUP THE STACK POINTER
                                ;;INITIALIZE A FEW VECTORS
004242  012737  051054  000020          MOV     #$SCOPE,@#IOTVEC ;;IOT VECTOR FOR SCOPE ROUTINE
004250  012737  000340  000022          MOV     #340,@#IOTVEC+2 ;;LEVEL 7
004256  012737  052710  000030          MOV     #$ERROR,@#EMTVEC ;;EMT VECTOR FOR ERROR ROUTINE
004264  012737  000340  000032          MOV     #340,@#EMTVEC+2 ;;LEVEL 7
004272  012737  053454  000034          MOV     #$TRAP,@#TRAPVEC ;;TRAP VECTOR FOR TRAP CALLS
004300  012737  000340  000036          MOV     #340,@#TRAPVEC+2;LEVEL 7
004306  012737  053524  000024          MOV     #$PWRDN,@#PWRVEC ;;POWER FAILURE VECTOR
004314  012737  000340  000026          MOV     #340,@#PWRVEC+2 ;;LEVEL 7
004322  005067  174664                  CLR     $TIMES          ;;INITIALIZE NUMBER OF ITERATIONS
004326  005067  174662                  CLR     $ESCAPE         ;;CLEAR THE ESCAPE ON ERROR ADDRESS
004332  112767  000001  174555          MOVB    #1,$ERMAX       ;;ALLOW ONE ERROR PER TEST
004340  012767  004340  174540          MOV     #.,$LPADR       ;;INITIALIZE THE LOOP ADDRESS FOR SCOPE
004346  012767  004346  174534          MOV     #.,$LPERR       ;;SETUP THE ERROR LOOP ADDRESS
                                ;;SIZE FOR A HARDWARE SWITCH REGISTER. IF NOT FOUND OR IT IS
                                ;;EQUAL TO A "-1", SETUP FOR A SOFTWARE SWITCH REGISTER.
004354  013746  000004                  MOV     @#ERRVEC,-(SP)  ;;SAVE ERROR VECTOR
004360  012737  004414  000004          MOV     #64$,@#ERRVEC   ;;SET UP ERROR VECTOR
004366  012767  177570  174544          MOV     #DSWR,SWR       ;;SETUP FOR A HARDWARE SWICH REGISTER
004374  012767  177570  174540          MOV     #DDISP,DISPLAY  ;;AND A HARDWARE DISPLAY REGISTER
004402  022777  177777  174530          CMP     #-1,@SWR        ;;TRY TO REFERENCE HARDWARE SWR
004410  001012                  BNE     66$                     ;;BRANCH IF NO TIMEOUT TRAP OCCURRED
                                                                ;;AND  THE HARDWARE SWR IS NOT = -1
004412  000403                  BR      65$                     ;;BRANCH IF NO TIMEOUT
004414  012716  004422  64$:    MOV     #65$,(SP)               ;;SET UP FOR TRAP RETURN
004420  000002                  RTI
004422  012767  000176  174510  65$:    MOV     #SWREG,SWR      ;;POINT TO SOFTWARE SWR
004430  012767  000174  174504          MOV     #DISPREG,DISPLAY
004436  012637  000004  66$:    MOV     (SP)+,@#ERRVEC  ;;RESTORE ERROR VECTOR


004442  012767  000000  173326          MOV     #0,PS                   ;SET PROCESSOR STATUS TO 0
004450  012777  051012  175140          MOV     #RPVECT,@RPVEC          ;THIS IS FOR UNTIMELY RP04 INTERRUPTS
004456  004737  052030                  JSR     PC,@#$TKINT     ;INITILIZE THE TK
004462  005737  004174                  TST     @#FIRST                 ;IS THIS FIRST TIME ROUND
004466  001001                  BNE     1$                      ;BRANCH IF NOT
004470  000402                  BR      2$
004472  000137  005306  1$:     JMP     @#SND1
004476                          2$:
004476  104400  004504                  TYPE    ,68$            ;;TYPE ASCIZ STRING
004502  000436                  BR      67$                     ;;GET OVER THE ASCIZ
```

```
                                        ;;68$:  .ASCIZ  <15><12>/RP04 DISKLESS CONTROLLER TEST-PART I (STATIC 1A)- DERPS-B/
004600                                  67$:
004600  104400  004606                        TYPE    ,70$            ;;TYPE ASCIZ STRING
004604  000425                              BR      69$             ;;GET OVER THE ASCIZ
                                        ;;70$:  .ASCIZ  <15><12>/MAKE SURE DCL IS LOCKED ON CORRECT PORT/
004660                                  69$:
004660  104400  004666                        TYPE    ,72$            ;;TYPE ASCIZ STRING
004664  000425                              BR      71$             ;;GET OVER THE ASCIZ
                                        ;;72$:  .ASCIZ  <15><12>/IF CHANGES ARE REQUIRED ON SWITCH THEN/
004740                                  71$:
004740  104400  004746                        TYPE    ,74$            ;;TYPE ASCIZ STRING
004744  000430                              BR      73$             ;;GET OVER THE ASCIZ
                                        ;;74$:  .ASCIZ  <15><12>/A CYCLE UP SEQUENCE IS REQUIRED FOR STROBING/
005026                                  73$:
005026  104400  005034                        TYPE    ,76$            ;;TYPE ASCIZ STRING
005032  000414                              BR      75$             ;;GET OVER THE ASCIZ
                                        ;;76$:  .ASCIZ  <15><12>/THE PORT SELECT FLOP/
005064                                  75$:
005064  104400  005072                        TYPE    ,78$            ;;TYPE ASCIZ STRING
005070  000432                              BR      77$             ;;GET OVER THE ASCIZ
                                        ;;78$:  .ASCIZ  <15><12>/ALL DCL UNDER TEST MUST BE LOCKED ON CORRECT PORT/
005156                                  77$:
005156  104400  005164                        TYPE    ,80$            ;;TYPE ASCIZ STRING
005162  000427                              BR      79$             ;;GET OVER THE ASCIZ
                                        ;;80$:  .ASCIZ  <15><12>/ALL DCL NOT UNDER TEST MUST BE SWITCHED OFF/
005242                                  79$:
005242  104400  005250                        TYPE    ,82$            ;;TYPE ASCIZ STRING
005246  000417                              BR      81$             ;;GET OVER THE ASCIZ
                                        ;;82$:  .ASCIZ  <15><12>/OR LOCKED ON THE OTHER PORT/
005306                                  81$:
005306  012737  177777  004174    SND1:   MOV     #-1,@#FIRST             ;NEXT TIME DO NOT GIVE HEADER
005314  005737  001766                    TST     @#SELECT                ;WAS IT A 200 START

005320  001435                            BEQ     TST1    ;BRANCH IF STARTING FROM 200

005322  104400  005330                    TYPE    ,65$            ;;TYPE ASCIZ STRING
005326  000423                            BR      64$             ;;GET OVER THE ASCIZ
                                        ;;65$:  .ASCIZ  <15><12>/SELECT UNIT NUMBER TO BE TESTED ?  /
005376                                  64$:
005376  104407                            RDOCT
005400  042716  177770                    BIC     #177770,(SP)            ;ONLY KEEP LAST 3 BITS
005404  011637  001760                    MOV     (SP),@#UNIT             ;SAVE UNIT TO BE TESTED
005410  012637  001770                    MOV     (SP)+,@#UNITSL          ;SAVE UNIT TO BE TESTED


                                        ;;*****************************************************************
                                        ;*TEST 1         REFERENCE EACH REGISTER
                                        ;*              REFERENCE EACH REGISTER BY A MOVE INSTRUCTION
                                        ;;*****************************************************************
005414  000004                    TST1:   SCOPE
005416  012767  000001  173566            MOV     #1,$TIMES       ;;DO 1 ITERATION
005424  012706  001000                    MOV     #STACK, SP      ;SET UP STACK POINTER

005430  012737  000001  004172            MOV     #TTNO,@#TSTNM   ;THIS SAVES TEST NUMBER
```

```
2938  005436  012737  052716  000030          MOV     #REGSA1,@#EMTVEC;ERROR VECTOR SO THAT
2939                                                                  ;NO REGISTERS ARE SAVED
2940  005444  012737  005472  000004          MOV     #2$,    @#ERRVEC ;SET UP FOR BUS TIMEOUT
2941  005452  012700  000024                  MOV     #24.,   R0      ;THERE ARE 24 REG TO TEST
2942  005456  012701  001620                  MOV     #RHDB,  R1      ;R1 NOW HAS ADDR OF ADDR OF FIRST REG.
2943  005462  013102                  1$:     MOV     @(R1)+, R2      ;READ HARDWARE REG.
2944  005464  005300                          DEC     R0              ;COUNT DOWN
2945  005466  001375                          BNE     1$              ;BRANCH IF 24 NOT DONE
2946  005470  000471                          BR      3$              ;BRANCH IF 24 DONE
2947  005472  012737  000006  000004  2$:     MOV     #ERRVEC+2,@#ERRVEC ;RESTORE TRAP CATCHER
2948  005500  022626                          CMP     (SP)+,  (SP)+   ;CLEAN STACK
2949  005502  016167  177776  173470          MOV     -2(R1), $TMP1   ;STORE FAILING REG ADDR
2950  005510  104015                          ERROR   15              ;REGISTER NON EXISTANT
2951  005512  032737  020000  001140          BIT     #SW13,@#SWR     ;INHIBIT ERROR PRINTOUT ?
2952  005520  001053                          BNE     4$              ;BRANCH IF YES
2953  005522  104400  005530                  TYPE    ,65$            ;;TYPE ASCIZ STRING
2954  005526  000431                          BR      64$             ;;GET OVER THE ASCIZ
2955                           ;;65$:  .ASCIZ  <15><12>/IF BASE ADDRESS IS TO BE CHANGED HALT PROGRAM /
2956  005612                   64$:
2957  005612  104400  005620                  TYPE    ,67$            ;;TYPE ASCIZ STRING
2958  005616  000411                          BR      66$             ;;GET OVER THE ASCIZ
2959                           ;;67$:  .ASCIZ  <15><12>/AND RESTART AT /
2960  005642                   66$:
2961  005642  012746  043452                  MOV     #BASECH,-(SP)   ;GET READY TO TYPE STARTING ADDRESS
2962                                                                  ;OF "CHANGE OF BASE ADDRESS" ROUTINE
2963  005646  104401                          TYPOC
2964  005650  000137  036564          4$:     JMP     @#$EOP          ;GO TO END OF PROGRAM
2965  005654  012737  052710  000030  3$:     MOV     #$ERROR,@#EMTVEC;RESTORE ERROR VECTOR
2966                                                                  ;SO THAT REGISTERS ARE SAVED
2967  005662  012737  000006  000004          MOV     #ERRVEC+2,@#ERRVEC ;RESTORE TRAP CATCHER
2968
2969
2970
2971
2972                           ;;**************************************************************
2973                           ;*TEST 2         RHCS2-CONTROL AND STATUS 2
2974
2975                           ;*      THIS PARTIALLY TESTS RHCS2 TO ENABLE DETERMINATION
2976                           ;*      OF THE NUMBER OF DRIVES PRESENT
2977
2978                           ;;**************************************************************
2979  005670  000004          TST2:   SCOPE
2980  005672  012767  000001  173312          MOV     #1,$TIMES       ;;DO 1 ITERATION
2981  005700  012706  001000                  MOV     #STACK,SP       ;RESET STACK
2982
2983
2984  005704  012737  000002  004172          MOV     #TTNO,@#TSTNM   ;THIS SAVES TEST NUMBER
2985
2986  005712  013737  010064  005732          MOV     @#PRCS2+12,@#UN
2987  005720  013737  001626  005734          MOV     @#RHCS2,@#UN+2
2988  005726  004537  037536                  JSR     R5,@#BITST      ;TEST BITS IN REGISTER
2989  005732  000000                  UN:     .WORD   0               ;ONLY THESE BITS ARE TEST READ/WRITE
2990  005734  000000                          .WORD   0               ;ADDRESS OF REG. BEING TESTED
2991  005736  104001                          ERROR   1               ;IN CORRECT DATA RECEIVED
2992  005740  000207                          RTS     PC              ;RETURN TO BLT3 ROUTINE
2993
```

```
                                        ;;*****************************************************************
                                        ;*TEST 3         PARTIAL TEST FOR RHAS FOR UNIT NUMBERS PRESENT
                                        ;;*****************************************************************
005742  000004                  TST3:   SCOPE
005744  012767  000001  173240          MOV     #1,$TIMES       ;;DO 1 ITERATION
005752  012737  000003  004172          MOV     #TTNO,@#TSTNM   ;THIS SAVES TEST NUMBER
005760  013701  001646                  MOV     @#RHAS,R1       ;R1 HAS ADDRESS OF RHAS
005764  012711  177777                  MOV     #-1,@R1 ;THIS CLEARS RHAS (SURPRISED!)
005770  011137  001126                  MOV     @R1,@#$BDDAT    ;TEST DATA
005774  105737  001126                  TSTB    @#$BDDAT
006000  001405                          BEQ     TST4    ;BRANCH IF GOOD
006002  005037  001124                  CLR     @#$GDDAT        ;GOOD DATA
006006  010137  037534                  MOV     R1,@#REGADR     ;FAILING REG. RHAS
006012  104001                          ERROR   1               ;RHAS DOES NOT CLEAR
                                                                ;WITH ONES
                                        ;;*****************************************************************
                                        ;*TEST 4         TEST FOR DRIVES PRESENT USING RHAS AND RHCS2
                                        ;;*****************************************************************
006014  000004                  TST4:   SCOPE
006016  012767  000001  173166          MOV     #1,$TIMES       ;;DO 1 ITERATION
006024  000005                          RESET                   ;START WITH AN INIT
006026  004737  052030                  JSR     PC,@#$TKINT     ;INITILIZE TK
006032  032737  020000  001140          BIT     #SW13,@#SWR     ;INHIBIT ERROR TYPEOUT?
006040  001030                          BNE     4$              ;BRANCH IF YES
006042  104400  006050                  TYPE    ,65$            ;;TYPE ASCIZ STRING
006046  000425                          BR      64$             ;;GET OVER THE ASCIZ
                                ;;65$:  .ASCIZ  <15><12>/LOOKING AT RHAS -  RP04 DRIVES PRESENT /
006122                          64$:
006122  013701  001646          4$:     MOV     @#RHAS,R1       ;R1 HAS ADDR. OF RHAS
006126  013702  001626                  MOV     @#RHCS2,R2      ;R2 HAS ADDR. OF RHCS2
006132  005012                          CLR     @R2             ;CLEAR RHCS2
006134  012700  000010                  MOV     #8.,R0          ;COUNT
006140  013704  001632                  MOV     @#RHER1,R4      ;R4 HAS ADDR. OF RHER1
006144  012714  177777          1$:     MOV     #-1,@R4         ;MOVE ERRORS INTO RHER1
006150  005212                          INC     @R2             ;INCREMENT UNIT NO.
006152  005300                          DEC     R0              ;COUNT
006154  001373                          BNE     1$              ;BRANCH IF 8 NOT DONE
006156  111137  002004                  MOVB    @R1,@#TOTALAT   ;SAVE TOTAL ATTENTION
                                                                ;USED IN DRIVE CLEAR TEST
006162  105037  002005                  CLRB    @#TOTALAT+1     ;CLEAR UPPER BYTE
006166  105711                          TSTB    @R1             ;TEST FOR ANY DRIVES PRESENT
006170  001402                          BEQ     2$              ;IF SOME  NOT THERE BRANCH
006172  000167  000436                  JMP     XE2             ;NONE THERE
006176  032737  020000  001140  2$:     BIT     #SW13,@#SWR     ;INHIBIT ERROR TYPE OUT?
006204  001402                          BEQ     3$              ;BRANCH IF NO
006206  000167  000706                  JMP     TST5    ;OUT
006212                          3$.
006212  104400  006220                  TYPE    ,67$            ;;TYPE ASCIZ STRING
006216  000412                          BR      66$             ;;GET OVER THE ASCIZ
                                ;;67$:  .ASCIZ  <15><12>/NO DRIVES-RHAS=0/
006244                          66$:
```

```
3050  006244  104400  006252                  TYPE    ,69$            ;;TYPE ASCIZ STRING
3051  006250  000436                          BR      68$             ;;GET OVER THE ASCIZ
3052                                  ;;69$:  .ASCIZ  <15><12>/WRITING ONES INTO ERROR REGISTER #1 FOR ALL UNIT NUMBERS/
3053  006346                          68$:
3054  006346  104400  006354                  TYPE    ,71$            ;;TYPE ASCIZ STRING
3055  006352  000441                          BR      70$             ;;GET OVER THE ASCIZ
3056                                  ;;71$:  .ASCIZ  <15><12>/DOES NOT SET ANY BIT IN THE ATTENTION REGISTER SO ABORT PROGRAM
3057  006456                          70$:
3058  006456  104400  006464                  TYPE    ,73$            ;;TYPE ASCIZ STRING
3059  006462  000440                          BR      72$             ;;GET OVER THE ASCIZ
3060                                  ;;73$:  .ASCIZ  <15><12>/TO LOOP ON THIS TEST WITHOUT PRINTOUT SET SWITCHS 13  8 AND 2/
3061  006564                          72$:
3062  006564  000137  036564                  JMP     @#$EOP          ;GO OUT
3063  006570  104400  006576                  TYPE    ,75$            ;;TYPE ASCIZ STRING
3064  006574  000410                          BR      74$             ;;GET OVER THE ASCIZ
3065                                  ;;75$:  .ASCIZ  <15><12>/TEST DRIVE 0/
3066  006616                          74$:
3067  006616  005037  001740                  CLR     @#UNITS
3068  006622  012767  000001  173132          MOV     #1,NOUNIT       ;NO. UNITS PRESENT=1
3069  006630  005037  001760                  CLR     @#UNIT
3070  006634                          XE2:
3071  006634  012700  000010          2$:     MOV     #8.,R0          ;COUNTER
3072  006640  012703  001740                  MOV     #UNITS,R3       ;POINTER
3073  006644  012723  177777          3$:     MOV     #-1,(R3)+       ;PRESET BLOCK TO ALL ONES
3074  006650  005300                          DEC     R0              ;COUNT
3075  006652  001374                          BNE     3$              ;BRANCH IF 8 NOT DONE
3076  006654  012703  001740                  MOV     #UNITS,R3       ;POINTER
3077  006660  005005                          CLR     R5
3078  006662  005037  001762                  CLR     @#NOUNIT        ;NO. OF UNITS PRESENT
3079  006666  012700  000010                  MOV     #8.,R0          ;COUNTER
3080  006672  011137  001176                  MOV     @R1,@#$TMP0     ;TEMPORARY STORAGE
3081  006676  006037  001176          4$:     ROR     @#$TMP0 ;SET CARRY IF ONE IN  0 BIT
3082
3083  006702  103065                          BCC     5$
3084  006704  010577  172716                  MOV     R5,@RHCS2       ;INSERT UNIT NUMBER
3085  006710  022777  024020  172736          CMP     #24020,@RHDT    ;IS THIS A DUAL PORT RP04
3086  006716  001450                          BEQ     6$              ;BRANCH IF YES
3087  006720  022777  020020  172726          CMP     #20020,@RHDT    ;IS THIS A SINGLE PORT RP04
3088  006726  001444                          BEQ     6$              ;BRANCH IF YES
3089  006730  104400  006736                  TYPE    ,65$            ;;TYPE ASCIZ STRING
3090  006734  000410                          BR      64$             ;;GET OVER THE ASCIZ
3091                                  ;;65$:  .ASCIZ  <15><12>/UNIT NUMBER /
3092  006756                          64$:
3093  006756  010546                          MOV     R5,-(SP)        ;GET READY TO TYPE UNIT NUMBER
3094  006760  104404                          TYPDS
3095  006762  104400  006770                  TYPE    ,67$            ;;TYPE ASCIZ STRING
3096  006766  000405                          BR      66$             ;;GET OVER THE ASCIZ
3097                                  ;;67$:  .ASCIZ  /,  RHDT= /
3098  007002                          66$:
3099  007002  017746  172646                  MOV     @RHDT,-(SP)     ;GET READY TO TYPE RHDT
3100  007006  104401                          TYPOC
3101  007010  104400  007016                  TYPE    ,69$            ;;TYPE ASCIZ STRING
3102  007014  000410                          BR      68$             ;;GET OVER THE ASCIZ
3103                                  ;;69$:  .ASCIZ  / ---NOT AN RP04/
3104  007036                          68$:
3105  007036  000407                          BR      5$              ;NO RP04 FOUND SO BRANCH
```

```
3106  007040  010523                  6$:     MOV     R5,(R3)+
3107  007042  104400  001223                  TYPE    ,$CRLF
3108  007046  010546                          MOV     R5,-(SP)
3109  007050  104404                          TYPDS                           ;TYPE DRIVE NO.
3110  007052  005237  001762                  INC     @#NOUNIT
3111  007056  005205                  5$:     INC     R5
3112  007060  005300                          DEC     R0
3113  007062  001305                          BNE     4$
3114  007064  013737  001740  001760          MOV     @#UNITS,@#UNIT
3115  007072  013737  001762  001764          MOV     @#NOUNIT,@#NUNIT        ;SAVE NO. OF UNITS
3116  007100  005337  001764                  DEC     @#NUNIT                 ;IF NUNIT = 0 THEN ONLY ONE UNIT
3117                                                                          ;IF NUNIT MORE THAN 0 THEN MORE THAN ONE UNIT
3118  007104  005737  001766                  TST     @#SELECT                ;STARTING ADDRESS 200 ?
3119
3120  007110  001403                          BEQ     TST5    ;BRANCH IF STARTING FROM 200
3121
3122  007112  013737  001770  001760          MOV     @#UNITSL,@#UNIT         ;SET UNIT NUMBER
3123
3124                                  ;;**********************************************************
3125                                  ;*TEST 5         TEST SERIAL NUMBER AND DRIVE TYPEI
3126                                  ;*              READ SERIAL NUMBER REGISTER AND DRIVE TYPE REGISTER
3127                                  ;*              TYPE IT OUT AND PROCEED
3128                                  ;*              TO LOOP HERE SET SWITCH 8 AND THIS TEST NO AND RESTART
3129
3130                                  ;;**********************************************************
3131  007120  000004                  TST5:   SCOPE
3132  007122  012767  000001  172062          MOV     #1,$TIMES       ;;DO 1 ITERATION
3133  007130  012767  007332  171750          MOV     #1$,$LPADR      ;;SET SCOPE LOOP ADDRESS
3134  007136  004737  040064                  JSR     PC,@#CLDISK             ;FILL UNIT NO.
3135  007142  005037  002002                  CLR     @#ATTENT                ;CLEAR
3136  007146  013700  001760                  MOV     @#UNIT,R0               ;R0 CONTAINS UNIT NO
3137  007152  116037  004176  002002          MOVB    ATABLE(R0),@#ATTENT     ;SET APPROPIATE ATTENTION BIT
3138  007160  104400  007166                  TYPE    ,65$            ;;TYPE ASCIZ STRING
3139  007164  000415                          BR      64$             ;;GET OVER THE ASCIZ
3140                                  ;;65$:  .ASCIZ  <15><12>/TESTING DRIVE NUMBER  /
3141  007220                          64$:
3142  007220  013746  001760                  MOV     @#UNIT,-(SP)            ;UNIT NO. TO STACK
3143  007224  104401                          TYPOC                           ;TYPE  DRIVE NO.
3144  007226  104400  001223                  TYPE    ,$CRLF
3145  007232  104400  007240                  TYPE    ,67$            ;;TYPE ASCIZ STRING
3146  007236  000410                          BR      66$             ;;GET OVER THE ASCIZ
3147                                  ;;67$:  .ASCIZ  <15><12>/SERIAL NO. = /
3148  007260                          66$:
3149  007260  017746  172372                  MOV     @RHSN,-(SP)     ;;SAVE @RHSN FOR TYPEOUT
3150  007264  104401                          TYPOC                   ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
3151  007266  104400  001223                  TYPE    ,$CRLF
3152  007272  104400  007300                  TYPE    ,69$            ;;TYPE ASCIZ STRING
3153  007276  000410                          BR      68$             ;;GET OVER THE ASCIZ
3154                                  ;;69$:  .ASCIZ  <15><12>/DRIVE TYPE = /
3155  007320                          68$:
3156  007320  017746  172330                  MOV     @RHDT,-(SP)     ;;SAVE @RHDT FOR TYPEOUT
3157  007324  104401                          TYPOC                   ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
3158  007326  104400  001223                  TYPE    ,$CRLF
3159  007332  005777  172320          1$:     TST     @RHSN           ;READ SERIAL NO. AND DRIVE TYPE
3160  007336  005777  172312                  TST     @RHDT           ;THESE TWO ARE TO HELP SCOPE LOOPS
3161  007342  017737  172310  001776          MOV     @RHSN,@#SAVSN   ;SAVE TO CHECK IF CLR RHCS2 BIT 5 CLEARS ANY BITS
```

```
3162  007350  017737  172300  001774          MOV     @RHDT,@#SAVDT   ;SAVE TO CHECK IF CLR RHCS2 BIT 5 CLEARS ANY BITS
3163
3164
3165
3166                                  ;;*****************************************************************
3167                                  ;*TEST 6         CHECK MOL TO BE LOW
3168
3169                                  ;*      MAKE SURE THAT DRIVE IS OFF LINE BEFORE STARTING PROGRAM
3170                                  ;*      IF DRIVE IS ON LINE THEN AFTER TYPE OUT THE PROGRAM WILL
3171                                  ;*      HANG FOR EVER WAITING FOR DRIVE TO GO OFF LINE
3172
3173                                  ;;*****************************************************************
3174  007356  000004                  TST6:   SCOPE
3175
3176  007360  012737  000006  004172          MOV     #TTNO,@#TSTNM   ;THIS SAVES TEST NUMBER
3177
3178  007366  004737  040064                  JSR     PC,@#CLDISK     ;GIVE INITILIZE
3179  007372  032713  010000                  BIT     #MOL,@R3        ;CHECK MOL IN RHDS1
3180
3181  007376  001550                          BEQ     TST7    ;BRANCH IF MOL LOW
3182
3183  007400  104400  007406                  TYPE    ,65$            ;;TYPE ASCIZ STRING
3184  007404  000421                          BR      64$             ;;GET OVER THE ASCIZ
3185                                  ;;65$:  .ASCIZ  <15><12>/DRIVE IS ON LINE - MOL IS HIGH/
3186  007450                          64$:
3187  007450  104400  007456                  TYPE    ,67$            ;;TYPE ASCIZ STRING
3188  007454  000424                          BR      66$             ;;GET OVER THE ASCIZ
3189                                  ;;67$:  .ASCIZ  <15><12>/HIT STOP ON DRIVE TO GET IT OFF LINE/
3190  007526                          66$:
3191  007526  104400  007534                  TYPE    ,69$            ;;TYPE ASCIZ STRING
3192  007532  000430                          BR      68$             ;;GET OVER THE ASCIZ
3193                                  ;;69$:  .ASCIZ  <15><12>/PROGRAM WILL HANG TESTING MOL TILL MOL IS LOW/
3194  007614                          68$:
3195  007614  032713  010000          1$:     BIT     #MOL,@R3        ;CHECK MOL IN RHDS1
3196  007620  001375                          BNE     1$              ;BRANCH IF MOL IS HIGH
3197  007622  104400  007630                  TYPE    ,71$            ;;TYPE ASCIZ STRING
3198  007626  000434                          BR      70$             ;;GET OVER THE ASCIZ
3199                                  ;;71$:  .ASCIZ  <15><12>/GOOD - MOL IS NOW LOW . PROGRAM WILL NOW BE EXECUTED/
3200  007720                          70$:
3201
3202
3203
3204                                  ;;*****************************************************************
3205                                  ;*TEST 7         RHWC - WORD COUNT REGISTER
3206                                  ;*              TEST LOADING AND READING OF ALL POSSIBLE BITS IN THE HARDWARE
3207                                  ;*              REGISTERS.  USE A PATTERN OF WALKING 0'S (-2,-3,-5 ETC.) AND
3208                                  ;*              WALKING 1'S (1,2,4,10 ETC)
3209                                  ;;*****************************************************************
3210  007720  000004                  TST7:   SCOPE
3211
3212
3213  007722  012706  001000                  MOV     #STACK,SP       ;RESET STACK
3214
3215
3216  007726  012737  000007  004172          MOV     #TTNO,@#TSTNM   ;THIS SAVES TEST NUMBER
3217
```

```
                                                                                
007734  013737  001622  007756          MOV     @#RHWC,@#PRWC+14        ;GET REGISTER ADDRESS
007742  013777  001760  171656  PRWC:   MOV     @#UNIT,@RHCS2   ;MOVE UNIT NO. UNDER TEST
007750  004567  027562                  JSR     R5,     BITST   ;TEST BITS IN REGISTER
007754  177777                          .WORD   177777          ;ONLY THESE BITS ARE TEST READ/WRITE
007756  176702                          .WORD   176702          ;ADDRESS OF REG. BEING TESTED
007760  104001                          ERROR   1               ;INCORRECT DATA RECEIVED
007762  000207                          RTS     PC              ;RETURN TO BLT3 ROUTINE
                                                                                
                                                                                
                                                                                
                                                                                
                                ;;**************************************************************
                                ;*TEST 10        RHBA - UNIBUS ADDRESS REGISTER
                                ;*              TEST LOADING AND READING OF ALL POSSIBLE BITS IN THE HARDWARE
                                ;*              REGISTERS.  USE A PATTERN OF WALKING 0'S (-2,-3,-5 ETC.) AND
                                ;*              WALKING 1'S (1,2,4,10 ETC)
                                ;;**************************************************************
007764  000004                TST10:  SCOPE
                                                                                
007766  012706  001000                  MOV     #STACK,SP       ;RESET STACK
                                                                                
007772  012737  000010  004172          MOV     #TTNO,@#TSTNM   ;THIS SAVES TEST NUMBER
                                                                                
010000  013737  001624  010022          MOV     @#RHBA,@#PRBA+14        ;GET REGISTER ADDRESS
010006  013777  001760  171612  PRBA:   MOV     @#UNIT,@RHCS2   ;MOVE UNIT NO. UNDER TEST
010014  004567  027516                  JSR     R5,     BITST   ;TEST BITS IN REGISTER
010020  177776                          .WORD   177776          ;ONLY THESE BITS ARE TEST READ/WRITE
010022  176704                          .WORD   176704          ;ADDRESS OF REG. BEING TESTED
010024  104001                          ERROR   1               ;INCORRECT DATA RECEIVED
010026  000207                          RTS     PC              ;RETURN TO BLT3 ROUTINE
                                                                                
                                                                                
                                                                                
                                                                                
                                ;;**************************************************************
                                ;*TEST 11        RHCS2 - CONTROL AND STATUS 2
                                ;*              TEST LOADING AND READING OF ALL POSSIBLE BITS IN THE HARDWARE
                                ;*              REGISTERS.  USE A PATTERN OF WALKING 0'S (-2,-3,-5 ETC.) AND
                                ;*              WALKING 1'S (1,2,4,10 ETC)
                                ;;**************************************************************
010030  000004                TST11:  SCOPE
                                                                                
010032  012706  001000                  MOV     #STACK,SP       ;RESET STACK
                                                                                
010036  012737  000011  004172          MOV     #TTNO,@#TSTNM   ;THIS SAVES TEST NUMBER
                                                                                
010044  013737  001626  010066          MOV     @#RHCS2,@#PRCS2+14      ;GET REGISTER ADDRESS
010052  013777  001760  171546  PRCS2:   MOV     @#UNIT,@RHCS2   ;MOVE UNIT NO. UNDER TEST
010060  004567  027452                  JSR     R5,     BITST   ;TEST BITS IN REGISTER
```

```
3274  010064  000017                        .WORD   17              ;ONLY THESE BITS ARE TEST READ/WRITE
3275  010066  176710                        .WORD   176710          ;ADDRESS OF REG. BEING TESTED
3276  010070  104001                        ERROR   1               ;INCORRECT DATA RECEIVED
3277  010072  000207                        RTS     PC              ;RETURN TO BLT3 ROUTINE
3278
3279
3280
3281
3282
3283                                ;;*****************************************************************
3284                                ;*TEST 12        RHCS1 - CONTROL AND STATUS 1 REGISTER
3285                                ;*              TEST LOADING AND READING OF ALL POSSIBLE BITS IN THE HARDWARE
3286                                ;*              REGISTERS.  USE A PATTERN OF WALKING 0'S (-2,-3,-5 ETC.) AND
3287                                ;*              WALKING 1'S (1,2,4,10 ETC)
3288                                ;;*****************************************************************
3289  010074  000004                TST12:  SCOPE
3290
3291
3292  010076  012706  001000                MOV     #STACK,SP       ;RESET STACK
3293
3294
3295  010102  012737  000012  004172        MOV     #TTNO,@#TSTNM   ;THIS SAVES TEST NUMBER
3296
3297
3298  010110  013737  001630  010132        MOV     @#RHCS1,@#PRCS1+14      ;GET REGISTER ADDRESS
3299  010116  013777  001760  171502  PRCS1: MOV     @#UNIT,@RHCS2   ;MOVE UNIT NO. UNDER TEST
3300  010124  004567  027406                JSR     R5,     BITST   ;TEST BITS IN REGISTER
3301  010130  001476                        .WORD   1476            ;ONLY THESE BITS ARE TEST READ/WRITE
3302  010132  176700                        .WORD   176700          ;ADDRESS OF REG. BEING TESTED
3303  010134  104001                        ERROR   1               ;INCORRECT DATA RECEIVED
3304  010136  000207                        RTS     PC              ;RETURN TO BLT3 ROUTINE
3305
3306
3307
3308
3309                                ;;*****************************************************************
3310                                ;*TEST 13        RHER1 - ERROR REGISTER #1
3311                                ;*              TEST LOADING AND READING OF ALL POSSIBLE BITS IN THE HARDWARE
3312                                ;*              REGISTERS.  USE A PATTERN OF WALKING 0'S (-2,-3,-5 ETC.) AND
3313                                ;*              WALKING 1'S (1,2,4,10 ETC)
3314                                ;;*****************************************************************
3315  010140  000004                TST13:  SCOPE
3316
3317
3318  010142  012706  001000                MOV     #STACK,SP       ;RESET STACK
3319
3320
3321  010146  012737  000013  004172        MOV     #TTNO,@#TSTNM   ;THIS SAVES TEST NUMBER
3322
3323
3324  010154  013737  001632  010176        MOV     @#RHER1,@#PRER1+14      ;GET REGISTER ADDRESS
3325  010162  013777  001760  171436  PRER1: MOV     @#UNIT,@RHCS2   ;MOVE UNIT NO. UNDER TEST
3326  010170  004567  027342                JSR     R5,     BITST   ;TEST BITS IN REGISTER
3327  010174  177777                        .WORD   177777          ;ONLY THESE BITS ARE TEST READ/WRITE
3328  010176  176714                        .WORD   176714          ;ADDRESS OF REG. BEING TESTED
3329  010200  104001                        ERROR   1               ;INCORRECT DATA RECEIVED
```

```
                                        RTS     PC              ;RETURN TO BLT3 ROUTINE


                                ;;*****************************************************************
                                ;*TEST 14       RHMR - MAINTENANCE REGISTER
                                ;*      BIT 0 (DMD) MUST BE SET BEFORE THE OTHER BITS
                                ;*      ARE READ WRITE
                                ;*      ONLY 5 LOW ORDER BITS ARE TESTED (R2 HAS 5)
                                ;;*****************************************************************
010204  000004                  TST14:  SCOPE


010206  012737  000014  004172          MOV     #TTNO,@#TSTNM   ;THIS SAVES TEST NUMBER

010214  004737  040064                  JSR     PC,@#CLDISK     ;SET UNIT NUMBER AND INIT
010220  013700  001650                  MOV     @#RHMR, R0      ;R0 HAS MAINTENANCE REG. ADR.
010224  012701  000001                  MOV     #1,     R1      ;R1 HAS DATA
010230  012702  000005                  MOV     #5,     R2      ;R2 HAS COUNT OF NUMBER OF BITS
010234  012710  000001          1$:     MOV     #DMD,   @R0     ;SET DIAGNOSTIC MODE BIT
010240  050110                  BIS     R1,     @R0     ;SET DATA IN RHMR
010242  010146                  MOV     R1,     -(SP)   ;SAVE DATA FOR COMPARES
010244  052716  000401                  BIS     #DMD!400,(SP);INCLUDE BIT 0
010250  011637  001124                  MOV     (SP),   @#$GDDAT ;SAVE FOR ERROR PRINTOUT
010254  022610                  CMP     (SP)+,  @R0     ;COMPARE DATA
010256  001405                  BEQ     2$              ;BRANCH IF GOOD
010260  011067  170642                  MOV     @R0,    $BDDAT  ;BAD DATA
010264  010037  037534                  MOV     R0,     @#REGADR ;FAILING REG. ADR.
010270  104001                  ERROR   1               ;MAINTENANCE REGISTER
                                                        ;FAILED TO SET INDICATED
                                                        ;BITS
010272  000241          2$:     CLC                     ;CLEAR CARRY
010274  006101                  ROL     R1              ;GET NEXT DATA
010276  052701  000400                  BIS     #400,R1 ;SET UNUSED BITS
010302  042701  001000                  BIC     #BIT09,R1       ;CLEAR READ ONLY BIT
010306  005302                  DEC     R2              ;COUNT
010310  001351                  BNE     1$              ;BRANCH IF 5 BITS NOT DONE


                                ;NOW FLOAT 0
010312  012701  000435                  MOV     #435,R1 ;R1 HAS DATA
010316  012702  000005                  MOV     #5,     R2      ;R2 HAS COUNT BITS
010322  012710  000001          3$:     MOV     #DMD,   @R0     ;SET DIAGNOSTIC MODE BITS
010326  050110                  BIS     R1,     @R0     ;SET DATA IN RHMR
010330  020110                  CMP     R1,     @R0     ;COMPARE DATA
010332  001407                  BEQ     4$              ;BRANCH IF GOOD
010334  010137  001124                  MOV     R1,     @#$GDDAT ;GOOD DATA
010340  011037  001126                  MOV     @R0,    @#$BDDAT ;BAD DATA
010344  010037  037534                  MOV     R0,     @#REGADR ;FAILING REG. ADR. RHMR
010350  104001                  ERROR   1               ;MAINTENANCE REGISTER
                                                        ;DOES NOT ALLOW WRITING
```

```
                                                        ;ZEROS
010352  000261                  4$:     SEC             ;SET CARRY
010354  006101                          ROL     R1      ;GET NEXT DATA
010356  042701  001340                  BIC     #BIT05!BIT06!BIT07!BIT09,R1     ;CLEAR READ ONLY BIT
010362  052701  000400                  BIS     #BIT08,R1       ;SET BIT ZEROED BY ROL
010366  005302                          DEC     R2      ;COUNT IF 5 BITS DONE
010370  001354                          BNE     3$      ;BRANCH IF INCOMPLETE


                        ;;*****************************************************************
                        ;*TEST 15       RHDST - DESIRED SECTOR/TRACK ADDRESS
                        ;*              TEST LOADING AND READING OF ALL POSSIBLE BITS IN THE HARDWARE
                        ;*              REGISTERS.  USE A PATTERN OF WALKING 0'S (-2,-3,-5 ETC.) AND
                        ;*              WALKING 1'S (1,2,4,10 ETC)
                        ;;*****************************************************************
010372  000004         TST15:  SCOPE


010374  012706  001000          MOV     #STACK,SP       ;RESET STACK

010400  012737  000015  004172  MOV     #TTNO,@#TSTNM   ;THIS SAVES TEST NUMBER

010406  013737  001634  010430  MOV     @#RHDST,@#PRDST+14      ;GET REGISTER ADDRESS
010414  013777  001760  171204  PRDST:  MOV     @#UNIT,@RHCS2   ;MOVE UNIT NO. UNDER TEST
010422  004567  027110          JSR     R5,     BITST   ;TEST BITS IN REGISTER
010426  017437                  .WORD   17437           ;ONLY THESE BITS ARE TEST READ/WRITE
010430  176706                  .WORD   176706          ;ADDRESS OF REG. BEING TESTED
010432  104001                  ERROR   1               ;INCORRECT DATA RECEIVED
010434  000207                  RTS     PC              ;RETURN TO BLT3 ROUTINE


                        ;;*****************************************************************
                        ;*TEST 16       RHER2 - ERROR REGISTER #2
                        ;*              TEST LOADING AND READING OF ALL POSSIBLE BITS IN THE HARDWARE
                        ;*              REGISTERS.  USE A PATTERN OF WALKING 0'S (-2,-3,-5 ETC.) AND
                        ;*              WALKING 1'S (1,2,4,10 ETC)
                        ;;*****************************************************************
010436  000004         TST16:  SCOPE


010440  012706  001000          MOV     #STACK,SP       ;RESET STACK

010444  012737  000016  004172  MOV     #TTNO,@#TSTNM   ;THIS SAVES TEST NUMBER

010452  013737  001636  010474  MOV     @#RHER2,@#PRER2+14      ;GET REGISTER ADDRESS
010460  013777  001760  171140  PRER2:  MOV     @#UNIT,@RHCS2   ;MOVE UNIT NO. UNDER TEST
010466  004567  027044          JSR     R5,     BITST   ;TEST BITS IN REGISTER
010472  177777                  .WORD   177777          ;ONLY THESE BITS ARE TEST READ/WRITE
```

```
3442  010474  176740                          .WORD   176740          ;ADDRESS OF REG. BEING TESTED
3443  010476  104001                          ERROR   1               ;INCORRECT DATA RECEIVED
3444  010500  000207                          RTS     PC              ;RETURN TO BLT3 ROUTINE
3445
3446
3447
3448
3449                                  ;;*****************************************************************
3450                                  ;*TEST 17        RHOF - MARGIN/OFFSET REGISTER
3451                                  ;*              TEST LOADING AND READING OF ALL POSSIBLE BITS IN THE HARDWARE
3452                                  ;*              REGISTERS.  USE A PATTERN OF WALKING 0'S (-2,-3,-5 ETC.) AND
3453                                  ;*              WALKING 1'S (1,2,4,10 ETC)
3454                                  ;;*****************************************************************
3455  010502  000004                  TST17:  SCOPE
3456
3457
3458  010504  012706  001000                  MOV     #STACK,SP       ;RESET STACK
3459
3460
3461  010510  012737  000017  004172          MOV     #TTNO,@#TSTNM   ;THIS SAVES TEST NUMBER
3462
3463
3464  010516  013737  001640  010540          MOV     @#RHOF,@#PROF+1         ;GET REGISTER ADDRESS
3465  010524  013777  001760  171074  PROF:   MOV     @#UNIT,@RHCS2   ;MOVE UNIT NO. UNDER TEST
3466  010532  004567  027000                  JSR     R5,     BITST   ;TEST BITS IN REGISTER
3467  010536  016277                          .WORD   16277           ;ONLY THESE BITS ARE TEST READ/WRITE
3468  010540  176732                          .WORD   176732          ;ADDRESS OF REG. BEING TESTED
3469  010542  104001                          ERROR   1               ;INCORRECT DATA RECEIVED
3470  010544  000207                          RTS     PC              ;RETURN TO BLT3 ROUTINE
3471
3472
3473
3474
3475                                  ;;*****************************************************************
3476                                  ;*TEST 20        RHCA - DESIRED CYLINDER REGISTER
3477                                  ;*              TEST LOADING AND READING OF ALL POSSIBLE BITS IN THE HARDWARE
3478                                  ;*              REGISTERS.  USE A PATTERN OF WALKING 0'S (-2,-3,-5 ETC.) AND
3479                                  ;*              WALKING 1'S (1,2,4,10 ETC)
3480                                  ;;*****************************************************************
3481  010546  000004                  TST20:  SCOPE
3482
3483
3484  010550  012706  001000                  MOV     #STACK,SP       ;RESET STACK
3485
3486
3487  010554  012737  000020  004172          MOV     #TTNO,@#TSTNM   ;THIS SAVES TEST NUMBER
3488
3489
3490  010562  013737  001642  010604          MOV     @#RHCA,@#PRCA+14        ;GET REGISTER ADDRESS
3491  010570  013777  001760  171030  PRCA:   MOV     @#UNIT,@RHCS2   ;MOVE UNIT NO. UNDER TEST
3492  010576  004567  026734                  JSR     R5,     BITST   ;TEST BITS IN REGISTER
3493  010602  001777                          .WORD   1777            ;ONLY THESE BITS ARE TEST READ/WRITE
3494  010604  176734                          .WORD   176734          ;ADDRESS OF REG. BEING TESTED
3495  010606  104001                          ERROR   1               ;INCORRECT DATA RECEIVED
3496  010610  000207                          RTS     PC              ;RETURN TO BLT3 ROUTINE
3497
```

```
                                ;;*****************************************************************
                                ;*TEST 21        RHER3 - ERROR REGISTER #3
                                ;*              TEST LOADING AND READING OF ALL POSSIBLE BITS IN THE HARDWARE
                                ;*              REGISTERS.  USE A PATTERN OF WALKING 0'S (-2,-3,-5 ETC.) AND
                                ;*              WALKING 1'S (1,2,4,10 ETC)
                                ;;*****************************************************************
010612  000004                TST21:  SCOPE

010614  012706  001000                MOV     #STACK,SP       ;RESET STACK

010620  012737  000021  004172        MOV     #TTNO,@#TSTNM   ;THIS SAVES TEST NUMBER

010626  013737  001644  010650        MOV     @#RHER3,@#PRER3+14      ;GET REGISTER ADDRESS
010634  013777  001760  170764 PRER3: MOV     @#UNIT,@RHCS2   ;MOVE UNIT NO. UNDER TEST
010642  004567  026670                JSR     R5,     BITST   ;TEST BITS IN REGISTER
010646  177777                        .WORD   177777          ;ONLY THESE BITS ARE TEST READ/WRITE
010650  176742                        .WORD   176742          ;ADDRESS OF REG. BEING TESTED
010652  104001                        ERROR   1               ;INCORRECT DATA RECEIVED
010654  000207                        RTS     PC              ;RETURN TO BLT3 ROUTINE


                                ;OF THE TWENTY REGISTERS (4 IN RH11, 16 IN RP04) ONLY 12 ARE
                                ;CHECKED IN THE ABOVE TESTS
                                ;TWO ARE ALREADY TESTED (SERIAL NO. AND DRIVE TYPE)
                                ;THE OTHER 7 WHICH ARE RHDS1, RHLA, RHCC, RHEC1, RHEC1, RHEC2
                                ;ARE READ ONLY REGISTERS.  ONE OR ZERO CANNOT BE WRITTEN



                                ;;*****************************************************************
                                ;*TEST 22        RHCS1 - BIT # 13 - MCPE
                                ;*       THIS FORCES A MASS BUS CONTROL PARITY ERROR
                                ;*       BY SETTING PAT AND READING RHER1
                                ;;*****************************************************************
010656  000004                TST22:  SCOPE
010660  012706  001000                MOV     #STACK,SP       ;RESET STACK

010664  012737  000022  004172        MOV     #TTNO,@#TSTNM   ;THIS SAVES TEST NUMBER

010672  004737  040064                JSR     PC,@#CLDISK     ;INIT AND SET UNIT NUMBER

                                ;SET FORCED PARITY ERROR BIT PAT
010676  052777  000020  170722        BIS     #PAT,@RHCS2     ;SET PAT TO INVERT PARITY
                                                        ;GENERATED
010704  005077  170722                CLR     @RHER1          ;WRITE DCL REGISTER
```

```
3554                                                    ;WITH THIS PARITY ERROR NOTHING WILL BE READ TILL IT IS
3555                                                    ;CLEARED
3556
3557  010710  011137  001126                  MOV     @R1,@#$BDDAT    ;TEST DATA
3558  010714  022737  104200  001126          CMP     #SC!DVA!RDY,@#$BDDAT ;COMPARE RHCS1 AFTER PARITY
3559                                                                    ;ERROR
3560  010722  001406                          BEQ     1$              ;BRANCH IF GOOD
3561  010724  012737  104200  001124          MOV     #SC!DVA!RDY,@#$GDDAT ;GOOD DATA
3562  010732  010137  037534                  MOV     R1,@#REGADR     ;REGISTER ADDRESS RHCS1
3563  010736  104001                          ERROR   1               ;SETTING PAT AND
3564                                                                    ;WRITING DCL REGISTER
3565                                                                    ;DID NOT SET SC,
3566                                                                    ;WITH PAT BIT HIGH
3567  010740  013746  001760          1$:     MOV     @#UNIT,-(SP)    ;GET UNIT NUMBER
3568  010744  052716  000120                  BIS     #PAT!IR,(SP)    ;INCLUDE PAT AND IR
3569  010750  012637  001124                  MOV     (SP)+,@#$GDDAT  ;
3570  010754  011237  001126                  MOV     @R2,@#$BDDAT    ;TEST DATA
3571  010760  023737  001124  001126          CMP     @#$GDDAT,@#$BDDAT       ;COMPARE RHCS2
3572  010766  001403                          BEQ     2$
3573  010770  010237  037534                  MOV     R2,@#REGADR     ;REGISTER ADDRESS
3574  010774  104001                          ERROR   1               ;READING DCL REGISTER
3575                                                                    ;WITH PAT HIGH CAUSED AN ERROR
3576
3577  010776  011437  001126          2$:     MOV     @R4,@#$BDDAT    ;TEST DATA
3578  011002  022737  000010  001126          CMP     #PAR,@#$BDDAT   ;ERROR REGISTER 1 SHOULD
3579                                                                    ;HAVE PAR SET
3580  011010  001406                          BEQ     3$
3581  011012  012737  000010  001124          MOV     #PAR,@#$GDDAT   ;GOOD DATA
3582  011020  010437  037534                  MOV     R4,@#REGADR     ;FAILING REGISTER RHER1
3583  011024  104001                          ERROR   1               ;PARITY ERROR DID NOT
3584                                                                    ;SET PAR
3585
3586  011026                          3$:
3587
3588
3589
3590
3591
3592
3593                                    ;;*****************************************************************
3594                                    ;*TEST 23       CONTROL AND STATUS 2 (RHCS 2)
3595                                    ;*      THIS TESTS THE UNIT SELECT BIT #0-2 (US1-4) CLEAR BIT #5 (CLR)
3596                                    ;*      NON-EXISTENT DRIVE BIT#12 (NED)
3597                                    ;*      THE OTHER RHCS2 BITS ARE NOT TESTED HERE
3598
3599                                    ;;*****************************************************************
3600  011026  000004                  TST23:  SCOPE
3601  011030  012706  001000                  MOV     #STACK,SP       ;RESET STACK
3602
3603
3604  011034  012737  000023  004172          MOV     #TTNO,@#TSTNM   ;THIS SAVES TEST NUMBER
3605
3606  011042  004737  040064                  JSR     PC,@#CLDISK                     ;HERE IT IS-USED TO LOAD CPU REGISTERS
3607                                                                  ;R1=RHCS1
3608                                                                  ;R2=RHCS2
3609                                                                  ;R3=RHDS1
```

```
3610  011046  005037  001772                    CLR     @#ERFLG$        ;CLEAR ERROR FLAG
3611                                    ;DISK TABLE WILL BE USED AS A TEMPORARY STORAGE
3612                                    ;FOR DRIVES PRESENT
3613  011052  012701  046330                    MOV     #DISK,R1        ;TABLE POINTER
3614  011056  012700  000010                    MOV     #8.,R0          ;MAX 8 DRIVES
3615  011062  012721  177777          1$:       MOV     #-1,(R1)+       ;FILL 8 LOCATIONS WITH -1
3616  011066  005300                            DEC     R0              ;COUNT
3617  011070  001374                            BNE     1$              ;BRANCH IF 8 NOT DONE
3618  011072  005012                            CLR     @R2             ;SELECT UNIT NO.0
3619  011074  012700  000010                    MOV     #8.,R0          ;MAX. 8 DRIVES
3620  011100  012701  046330                    MOV     #DISK,R1        ;TABLE POINTER
3621  011104  005714                  2$:       TST     @R4             ;DO A READ ON RHER1(ANY REG)
3622  011106  032712  010000                    BIT     #NED,@R2        ;TEST NON EXISTENT DRIVE
3623  011112  001415                            BEQ     3$              ;IF DRIVE PRESENT BRANCH
3624  011114  005300                  7$:       DEC     R0              ;DECREMENT DRIVE COUNT
3625  011116  001434                            BEQ     4$              ;BRANCH IF FINISHED
3626  011120  011246                            MOV     @R2,-(SP)       ;GET RHCS2 FOR UNIT NUMBER
3627  011122  042716  177770                    BIC     #^C7,(SP)       ;KEEP ONLY UNIT NUMBER
3628  011126  005216                            INC     (SP)            ;INCREMENT UNIT UNMBER
3629  011130  013703  001630                    MOV     @#RHCS1,R3      ;GET RHCS1 ADDRESS
3630  011134  005203                            INC     R3              ;GET UPPER BYTE OF RHCS1
3631  011136  112713  000100                    MOVB    #100,@R3        ;SET TRE IN RHCS1
3632                                            ;WITHOUT ADDRESSING DRIVE
3633  011142  012612                            MOV     (SP)+,@R2       ;RHCS2 HAS INCREMENTED UNIT WITH NED CLEARED
3634  011144  000757                            BR      2$              ;GO TO TEST ANY REG. HERE RHER1
3635  011146  022777  024020  170500  3$:       CMP     #24020,@RHDT    ;IS THIS A DUAL PORT RP04
3636  011154  001405                            BEQ     8$              ;BRANCH IF YES
3637  011156  022777  020020  170470            CMP     #20020,@RHDT    ;IS THIS A SINGLE PORT RP04
3638  011164  001401                            BEQ     8$              ;BRANCH IF YES
3639  011166  000752                            BR      7$              ;NO RP04 FOUND SO BRANCH
3640  011170  012746  000010          8$:       MOV     #8.,-(SP)       ;GETTIN READY TO TYPE PRESENT DRIVES
3641  011174  160016                            SUB     R0,(SP)         ;(SP) NOW HAS DRIVE PRESENT NO.
3642  011176  012621                            MOV     (SP)+,(R1)+     ;SAVE UNIT NO.
3643  011200  005300                            DEC     R0              ;DECREMENT DRIVE COUNT
3644  011202  001402                            BEQ     4$              ;BRANCH IF FINISHED
3645  011204  005212                            INC     @R2             ;SELECT NEXT UNIT
3646  011206  000736                            BR      2$
3647  011210  004037  040736          4$:       JSR     R0,@#COMPAR     ;CHECK
3648  011214  001740                            UNITS                   ;GOOD DATA
3649  011216  046330                            DISK                    ;TEST DATA
3650  011220  000010                            8.                      ;NO.OF WORDS
3651  011222  011230                            5$                      ;RETURN FOR ERROR HEADER
3652  011224  011234                            6$                      ;RETURN FOR ERROR DATA
3653  011226  011240                            NEXT1                   ;RETURN FOR GOOD COMPARISON
3654  011230  104022                  5$:       ERROR   22
3655  011232  000207                            RTS     PC              ;RETURN TO COMPARISON ROUTINE
3656  011234  104023                  6$:       ERROR   23              ;UNIT NUMBER FOUND BY SETTING RHAS
3657  011236  000207                            RTS     PC              ;AND UNIT UNMBER FOUND BY CHECKING
3658                                            ;RHCS2 NED BIT #12 DID NOT AGREE

        ;/#\:/#\:/#\:/#\:/#\:/#\:/#\:/#\:/#\:/#\:/#\:/#\:/#\:/#\:/#\:/#\:/#\:/#\:/#\:/#\:
        ;/#\:/#\:/#\:/#\:/#\:/#\:/#\:/#\:/#\:/#\:/#\:/#\:/#\:/#\:/#\:/#\:/#\:/#\:/#\:/#\:
        ;/#\:/#\:/#\:/#\:/#\:/#\:/#\:/#\:/#\:/#\:/#\:/#\:/#\:/#\:/#\:/#\:/#\:/#\:/#\:/#\:
        ;IN THE ABOVE TEST BITS 0,1,2, AND BIT 12 ARE TESTED
        ;IF THE "DRIVES PRESENT" TYPE OUT DO NOT AGREE WITH WHAT IS
        ;REALLY THERE THEN THE ERROR IS IN THE LOGIC
```

```
                ;FOR BIT12(NED) OR UNIT SELECT(BIT 0 TO 2)
                ;IT IS NOT POSSIBLE BY PROGRAM TO CHECK IF A NON-EXISTENT
                ;DRIVE IS REALLY STANDING THERE OR NOT
                ;MANUALLY LOAD LOCATION "ERUNIT" WITH A UNIT NUMBER
                ;AND RESTART AT LOCATION "ERSTAR" THIS WILL LOOP FOR
                ;EVER DOING EXACTLY AS TEST ON THAT ONE UNIT
                ;TO GET BACK TO MAIN DIAGNOSTIC HIT HALT SWITCH AND
                ;RESTART PROGRAM IN NORMAL MANNER
                ;/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*
                ;/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*
                ;/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*

3678
3679
3680
3681                                    ;THE FOLLOWING TEST CHECKS CLR (RHCS2 BIT #5)
3682                                    ;ALL REGISTERS ARE LOADED WITH ALL ONES EXCEPT BIT #0 AND #6
3683                                    ;WHICH ARE "GO" AND "INTERRUPT ENABLE"
3684                                    ;THEN CLR IS GIVEN
3685                                    ;RHDB IS READ FIRST AS THIS WILL SET DTL IN RHCS2 AND
3686                                    ;SC AND TRE IN RHCS1 ANOTHER CLR IS GIVEN THEN ALL OTHER REGISTERS
3687                                    ;ARE READ
3688
3689                                    .
3690  011240  012737  011240  001110  NEXT1:  MOV     #NEXT1,@#$LPERR ;SET LOOP POINT TO NEXT1
3691  011246  004737  040064                  JSR     PC,@#CLDISK     ;SET REGISTERS AND CLEAR
3692                                          ;FILL ALL POSSIBLE BITS WITH ONES
3693  011252  012777  177777  170340          MOV     #177777,@RHDB   ;BUS ADDRESS REGISTER GETS 177777
3694  011260  012777  177777  170334          MOV     #177777,@RHWC   ;WORD COUNT REGISTER GETS 177777
3695  011266  012777  177777  170330          MOV     #177777,@RHBA   ;BUS ADDRESS REGISTER GETS 177777
3696  011274  052777  157010  170324          BIS     #157010,@RHCS2  ;CONTROL AND STATUS 2 GETS 157010
3697  011302  012777  001476  170320          MOV     #1476,@RHCS1    ;CONTROL AND STATUS REGISTER/GETS 1476
3698  011310  012777  177777  170314          MOV     #177777,@RHER1  ;ERROR REGISTER1 GETS 177777
3699  011316  012777  017437  170310          MOV     #17437,@RHDST   ;DESIRED SECTOR TRACK
3700  011324  012777  177777  170304          MOV     #177777,@RHER2  ;ERROR REGISTER 2
3701  011332  012777  016277  170300          MOV     #16277,@RHOF    ;OFFSET REGISTER
3702  011340  012777  177777  170274          MOV     #177777,@RHCA   ;DESIRED CYLINDER
3703  011346  012777  177777  170270          MOV     #177777,@RHER3  ;ERROR REGISTER 3
3704  011354  012777  000001  170266          MOV     #DMD,@RHMR      ;MAINTENANCE REGISTER
3705  011362  012777  177777  170260          MOV     #177777,@RHMR   ;MAINTENANCE REGISTER
3706
3707
3708
3709  011370  052712  000040                  BIS     #CLR,   @R2     ;CLEAR ALL POSSIBLE BITS
3710  011374  013712  001760                  MOV     @#UNIT,@R2      ;REINSTATE UNIT NO.
3711  011400  012700  001620                  MOV     #RHDB,  R0      ;R0 CONTAINS ADDR. OF ADDR. OF REG.
3712
3713                                          ;DATA BUFFER REGISTER
3714
3715  011404  012737  177777  001124          MOV     #177777,@#$GDDAT;GOOD DATA FOR ERROR
3716  011412  011037  037534                  MOV     @R0,@#REGADR    ;REGISTER ADDRESS
3717  011416  013037  001126                  MOV     @(R0)+,@#$BDDAT ;TEST DATA
3718  011422  023737  001124  001126          CMP     @#$GDDAT,@#$BDDAT;COMPARE GOOD WITH TEST DATA
3719  011430  001401                          BEQ     2$              ;BRANCH IF GOOD
3720  011432  104001                          ERROR   1               ;RHDB DID NOT HAVE ALL ONES
3721                                                                  ;AFTER A CLR IN RHCS2
```

```
3722
3723  011434  052712  000040          2$:   BIS     #CLR,@R2        ;SET CLEAR AGAIN BECAUSE
3724                                          ;READING RHDB AFTER CLEARING  WILL
3725                                          ;SET DLT SC AND TRE
3726  011440  013712  001760                MOV     @#UNIT,@R2      ;REINSTATE UNIT NO.
3727
3728
3729                                  ;WORD COUNT REGISTER
3730
3731
3732  011444  012737  177777  001124  3$:   MOV     #177777,        @#$GDDAT ;GOOD DATA FOR ERROR TYPEOUT
3733  011452  013037  001126          MOV     @(R0)+,@#$BDDAT ;TEST DATA
3734  011456  022737  177777  001126  CMP     #177777,        @#$BDDAT;COMPARE DATA
3735  011464  001402                  BEQ     4$              ;BRANCH IF GOOD
3736  011466  004737  012252          JSR     PC,@#ERCS2C     ;JUMP TO ERROR FOR CLR (BIT 5)
3737                                          ;IN RHCS2
3738
3739
3740                                  ;BUS ADDRESS REGISTER
3741
3742
3743  011472  012737  000000  001124  4$:   MOV     #0,     @#$GDDAT ;GOOD DATA FOR ERROR TYPEOUT
3744  011500  013037  001126          MOV     @(R0)+,@#$BDDAT ;TEST DATA
3745  011504  022737  000000  001126  CMP     #0,     @#$BDDAT;COMPARE DATA
3746  011512  001402                  BEQ     5$              ;BRANCH IF GOOD
3747  011514  004737  012252          JSR     PC,@#ERCS2C     ;JUMP TO ERROR FOR CLR (BIT 5)
3748                                          ;IN RHCS2
3749
3750
3751                                  ;CONTROL AND STATUS 2 REGISTER
3752
3753
3754
3755  011520  012746  000100          5$:   MOV     #100,   -(SP)   ;INCLUDE IR
3756  011524  053716  001760          BIS     @#UNIT, (SP)    ;SET UNIT NO.
3757  011530  012637  001124          MOV     (SP)+,  @#$GDDAT;GOOD DATA FOR TYPE OUT
3758  011534  013037  001126          MOV     @(R0)+,@#$BDDAT ;TEST DATA
3759  011540  023737  001124  001126  CMP     @#$GDDAT,@#$BDDAT;COMPARE DATA
3760  011546  001402                  BEQ     6$              ;BRANCH IF GOOD
3761  011550  004737  012252          JSR     PC,@#ERCS2C     ;JUMP TO ERROR FOR CLR (BIT 5)
3762                                          ;IN RHCS2
3763
3764
3765
3766                                  ;CONTROL AND STATUS 1 REGISTER
3767
3768
3769  011554  012737  004276  001124  6$:   MOV     #4276,  @#$GDDAT ;GOOD DATA FOR ERROR TYPEOUT
3770  011562  013037  001126          MOV     @(R0)+,@#$BDDAT ;TEST DATA
3771  011566  022737  004276  001126  CMP     #4276,  @#$BDDAT;COMPARE DATA
3772  011574  001402                  BEQ     7$              ;BRANCH IF GOOD
3773  011576  004737  012252          JSR     PC,@#ERCS2C     ;JUMP TO ERROR FOR CLR (BIT 5)
3774                                          ;IN RHCS2
3775
3776                                  ;ERROR 1 REGISTER
3777
```

```
3778
3779  011602  012737  000000  001124  7$:     MOV     #0,     @#$GDDAT ;GOOD DATA FOR ERROR TYPEOUT
3780  011610  013037  001126                  MOV     @(R0)+,@#$BDDAT ;TEST DATA
3781  011614  022737  000000  001126          CMP     #0,     @#$BDDAT;COMPARE DATA
3782  011622  001402                          BEQ     10$             ;BRANCH IF GOOD
3783  011624  004737  012252                  JSR     PC,@#ERCS2C     ;JUMP TO ERROR FOR CLR (BIT 5)
3784                                                                  ;IN RHCS2
3785
3786                                          ;DESIRED SECTOR/TRACK REGISTER
3787
3788
3789  011630  012737  017437  001124  10$:    MOV     #17437, @#$GDDAT ;GOOD DATA FOR ERROR TYPEOUT
3790  011636  013037  001126                  MOV     @(R0)+,@#$BDDAT ;TEST DATA
3791  011642  022737  017437  001126          CMP     #17437, @#$BDDAT;COMPARE DATA
3792  011650  001402                          BEQ     11$             ;BRANCH IF GOOD
3793  011652  004737  012252                  JSR     PC,@#ERCS2C     ;JUMP TO ERROR FOR CLR (BIT 5)
3794                                                                  ;IN RHCS2
3795
3796                                          ;ERROR 2 REGISTER
3797
3798
3799  011656  012737  000000  001124  11$:    MOV     #0,     @#$GDDAT ;GOOD DATA FOR ERROR TYPEOUT
3800  011664  013037  001126                  MOV     @(R0)+,@#$BDDAT ;TEST DATA
3801  011670  022737  000000  001126          CMP     #0,     @#$BDDAT;COMPARE DATA
3802  011676  001402                          BEQ     12$             ;BRANCH IF GOOD
3803  011700  004737  012252                  JSR     PC,@#ERCS2C     ;JUMP TO ERROR FOR CLR (BIT 5)
3804                                                                  ;IN RHCS2
3805
3806                                          ;OFFSET REGISTER
3807
3808
3809  011704  012737  116000  001124  12$:    MOV     #116000,        @#$GDDAT ;GOOD DATA FOR ERROR TYPEOUT
3810  011712  013037  001126                  MOV     @(R0)+,@#$BDDAT ;TEST DATA
3811  011716  022737  116000  001126          CMP     #116000,        @#$BDDAT;COMPARE DATA
3812  011724  001402                          BEQ     13$             ;BRANCH IF GOOD
3813  011726  004737  012252                  JSR     PC,@#ERCS2C     ;JUMP TO ERROR FOR CLR (BIT 5)
3814                                                                  ;IN RHCS2
3815
3816                                          ;DESIRED CYLINDER ADDRESS REGISTER
3817
3818
3819  011732  012737  001777  001124  13$:    MOV     #1777,  @#$GDDAT ;GOOD DATA FOR ERROR TYPEOUT
3820  011740  013037  001126                  MOV     @(R0)+,@#$BDDAT ;TEST DATA
3821  011744  022737  001777  001126          CMP     #1777,  @#$BDDAT;COMPARE DATA
3822  011752  001402                          BEQ     14$             ;BRANCH IF GOOD
3823  011754  004737  012252                  JSR     PC,@#ERCS2C     ;JUMP TO ERROR FOR CLR (BIT 5)
3824                                                                  ;IN RHCS2
3825
3826                                          ;ERROR 3 REGISTER
3827
3828
3829  011760  012737  000000  001124  14$:    MOV     #0,     @#$GDDAT ;GOOD DATA FOR ERROR TYPEOUT
3830  011766  013037  001126                  MOV     @(R0)+,@#$BDDAT ;TEST DATA
3831  011772  022737  000000  001126          CMP     #0,     @#$BDDAT;COMPARE DATA
3832  012000  001402                          BEQ     15$             ;BRANCH IF GOOD
3833  012002  004737  012252                  JSR     PC,@#ERCS2C     ;JUMP TO ERROR FOR CLR (BIT 5)
```

```
                                                        ;IN RHCS2
                                
                                        ;ATTENTION SUMMARY REGISTER
                                
012006  013037  001126          15$:  MOV   @(R0)+,@#$BDDAT     ;GET RHAS CONTENTS
012012  012737  000000  001124        MOV   #0,@#$GDDAT         ;GOOD DATA FOR ERROR TYPE OUT
012020  123737  001124  001126        CMPB  @#$GDDAT,@#$BDDAT   ;COMPARE FOR RHAS
012026  001402                      BEQ   16$                 ;BRANCH IF GOOD
012030  004737  012252              JSR   PC,@#ERCS2C         ;JUMP TO ERROR FOR CLR (BIT 5)
                                                              ;IN RHCS2
                                
                                        ;MAINTAINABILITY REGISTER
                                
012034  012737  000400  001124  16$:  MOV   #400,   @#$GDDAT ;GOOD DATA FOR ERROR TYPEOUT
012042  013037  001126              MOV   @(R0)+,@#$BDDAT ;TEST DATA
012046  022737  000400  001126      CMP   #400,   @#$BDDAT;COMPARE DATA
012054  001402                      BEQ   17$                 ;BRANCH IF GOOD
012056  004737  012252              JSR   PC,@#ERCS2C         ;JUMP TO ERROR FOR CLR (BIT 5)
                                                              ;IN RHCS2
                                
                                        ;DRIVE STATUS REGISTER
                                
012062  012737  000600  001124  17$:  MOV   #600,@#$GDDAT     ;GOOD DATA FOR ERROR TYPEOUT
012070  013046                      MOV   @(R0)+,-(SP)        ;GET RHDS1
012072  011637  001126              MOV   (SP),@#$BDDAT       ;TEST DATA
012076  042716  001100              BIC   #VV!PROG,(SP)       ;CLEAR VV AND PROG
012102  022726  000600              CMP   #600,(SP)+          ;COMPARE DATA
012106  001402                      BEQ   20$                 ;BRANCH IF GOOD
012110  004737  012252              JSR   PC,@#ERCS2C         ;JUMP TO ERROR FOR CLR (BIT 5)
                                                              ;IN RHCS2
                                
                                        ;DRIVE TYPE
                                
012114  013737  001774  001124  20$:  MOV   @#SAVDT,          @#$GDDAT ;GOOD DATA FOR ERROR TYPEOUT
012122  013037  001126              MOV   @(R0)+,@#$BDDAT ;TEST DATA
012126  023737  001774  001126      CMP   @#SAVDT,          @#$BDDAT;COMPARE DATA
012134  001402                      BEQ   21$                 ;BRANCH IF GOOD
012136  004737  012252              JSR   PC,@#ERCS2C         ;JUMP TO ERROR FOR CLR (BIT 5)
                                                              ;IN RHCS2
                                
                                        ;SERIAL NUMBER REGISTER
                                
012142  013737  001776  001124  21$:  MOV   @#SAVSN,          @#$GDDAT ;GOOD DATA FOR ERROR TYPEOUT
012150  013037  001126              MOV   @(R0)+,@#$BDDAT ;TEST DATA
012154  023737  001776  001126      CMP   @#SAVSN,          @#$BDDAT;COMPARE DATA
012162  001402                      BEQ   22$                 ;BRANCH IF GOOD
012164  004737  012252              JSR   PC,@#ERCS2C         ;JUMP TO ERROR FOR CLR (BIT 5)
                                                              ;IN RHCS2
```

```
                                        ;ECC1 POSITION

012170  012737  000000  001124  22$:    MOV     #0,     @#$GDDAT ;GOOD DATA FOR ERROR TYPEOUT
012176  013037  001126                  MOV     @(R0)+,@#$BDDAT ;TEST DATA
012202  022737  000000  001126          CMP     #0,     @#$BDDAT;COMPARE DATA
012210  001402                          BEQ     23$             ;BRANCH IF GOOD
012212  004737  012252                  JSR     PC,@#ERCS2C     ;JUMP TO ERROR FOR CLR (BIT 5)
                                                                ;IN RHCS2

                                        ;ECC2 PATTERN

012216  012737  000000  001124  23$:    MOV     #0,     @#$GDDAT ;GOOD DATA FOR ERROR TYPEOUT
012224  013037  001126                  MOV     @(R0)+,@#$BDDAT ;TEST DATA
012230  022737  000000  001126          CMP     #0,     @#$BDDAT;COMPARE DATA
012236  001402                          BEQ     24$             ;BRANCH IF GOOD
012240  004737  012252                  JSR     PC,@#ERCS2C     ;JUMP TO ERROR FOR CLR (BIT 5)
                                                                ;IN RHCS2

                                        ;LOOK-AHEAD REGISTER

012244  005720          24$:    TST     (R0)+           ;AS THE LOOK-AHEAD REG. CANNOT BE PREDICTED
                                                        ;AFTER AN INIT IT IS NOT CHECKED

                                        ;CURRENT CYLINDER ADDRESS REGISTER

012246  005720          25$:    TST     (R0)+           ;AS THE CURRENT REG. CANNOT BE PREDICTED
                                                        ;AFTER A INIT IT IS NOT CHECKED

012250          26$:

012250  000405                  BR      TST24   ;BRANCH OVER JSR


012252  014037  037534  ERCS2C: MOV     -(R0),  @#REGADR ;FAILING REGISTER ADDRESS
012256  104001                  ERROR   1               ;CLR (BIT 5) IN RHCS2 DID
                                                        ;NOT CLEAR APPROPIATE BITS
                                                        ;OR CLEARED EXTRA BITS
012260  005720                  TST     (R0)+           ;UNDO -(R0) FOR BAD DATA
012262  000207                  RTS     PC              ;RETURN TO TEST ABOVE

                                .SBTTL  SILO TEST


                        ;;**************************************************************
                        ;*TEST 24        SILO TST 1

                        ;*      THIS TESTS THE SILO BUFFER IN THE RH11 CONTROLLER
                        ;*      A READ IS ATTEMPTED FROM AN EMPTY SILO
                        ;*      DATA LATE (DLT) (RHCS2), TRANSFER ERROR (TRE) (RHCS1),
```

```
                                        ;*      SPECIAL CONDITION (SC) (RHCS1) SHOULD SET
                                        ;*      THEN LOADING "1" INTO TRE SHOULD CLEAR DLT, TRE AND SC
                                        ;
                                        ;;*****************************************************************
012264  000004                          TST24:  SCOPE
012266  012706  001000                          MOV     #STACK,SP       ;RESET STACK
012272  012737  000024  004172                  MOV     #TTNO,@#TSTNM   ;THIS SAVES TEST NUMBER
012300  000167  000130                          JMP     TST25   ;JUMP TO NEXT TEST FOR RH70
012304  004767  025554                          JSR     PC,CLDISK       ;CLEAR DISK AND LOAD R'S
012310  017700  167304                          MOV     @RHDB,R0        ;READ FROM EMPTY SILO
012314  013746  001760                          MOV     @#UNIT,-(SP)    ;GET UNIT NO. IN
012320  052716  100100                          BIS     #DLT!IR,(SP)    ;GET DATA LATE BIT AND IR
012324  004737  037470                          JSR     PC,@#PUTREG     ;SAVE REGISTERS
012330  022637  001676                          CMP     (SP)+,@#CS2     ;IS DATA LATE BIT UP?
012334  001403                              BEQ     1$              ;IF YES BRANCH
012336  010237  001122                          MOV     R2,@#$BDADR     ;IF NOT STORE FAILING REG.
012342  104011                              ERROR   11              ;RHCS2 DID NOT HAVE DLT
                                                                    ;RHCS2 SHOULD HAVE ONLY
                                                                    ;DLT AND UNIT NUMBER (BIT 0-2)
                                                                    ;ALL OTHER BITS SHOULD
                                                                    ;BE 0
012344  022737  144200  001700  1$:             CMP     #SC!TRE!RDY!DVA,@#CS1 ;IS SPECIAL CONDITION, TRANSFER ERROR
                                                                    ;AND READY UP?
012352  001403                              BEQ     2$              ;IF YES BRANCH
012354  010137  001122                          MOV     R1,@#$BDADR     ;IF NOT STORE FAILING REG.
012360  104011                              ERROR   11              ;RHCS1 DID NOT HAVE SC, DVA
                                                                    ;TRE AND RDY. AFTER A
                                                                    ;READ FROM EMPTY SILO ONLY
                                                                    ;THESE BITS SHOULD BE UP
                                                                    ;ALL OTHERS SHOULD BE 0
012362  012711  040000              2$:         MOV     #TRE,@R1        ;CLEAR ERROR BITS BY MOVING
                                                                    ;ONE INTO TRE IN RHCS1
012366  004737  037470                          JSR     PC,@#PUTREG     ;SAVE REGISTERS
012372  022737  004200  001700                  CMP     #RDY!DVA,@#CS1  ;ALL BITS BUT RDY AND DVA SHOULD
                                                                    ;BE 0
012400  001403                              BEQ     3$              ;BRANCH IF YES
012402  010137  001122                          MOV     R1,@#$BDADR     ;STORE FAILING ADDRESS
012406  104011                              ERROR   11              ;AFTER A ONE IN TRE ONLY
                                                                    ;READY AND DVA SHOULD BE SET IN
                                                                    ;RHCS1
012410  013746  001760              3$:         MOV     @#UNIT,-(SP)
012414  052716  000100                          BIS     #IR,(SP)
012420  022637  001676                          CMP     (SP)+,@#CS2     ;RHCS2 SHOULD HAVE IR AND UNIT ONLY
012424  001403                              BEQ     TST25   ;BRANCH IF YES
012426  010237  001122                          MOV     R2,@#$BDADR     ;STORE FAILING ADDR
012432  104011                              ERROR   11              ;AFTER A ONE IN TRE ONLY
                                        ;;*****************************************************************
                                        ;*TEST 25       SILO TEST 2
                                        ;*      THIS TESTS THE IR AND "OR" BITS OF RHCS2
                                        ;*      AT THE BEGINNING IR SHOULD BE SET AND "OR" RESET
```

```
                                        ;*      LOADING 0 IN SILO RESETS IR FOR ONLY 2 MICRO SECONDS
                                        ;*      THIS TIME CANNOT BE CHECKED BUT IT IS CHECKED TO SEE IF
                                        ;*      IT DOES GO DOWN OR NOT
                                        ;*      THEN ALL 1 IS LOADED IN SILO "OR" SHOULD BECOME SET
                                        ;*      IN 30 MICRO SECONDS AGAIN TIME IS NOT  CHECKED
                                        ;*      "OR" SHOULD BE SET
                                        ;*      THE OUTPUT FROM THE SILO SHOULD BE 0 AND ALL ONES

012434  000004                         ;;*****************************************************************
                                TST25:  SCOPE
012436  012706  001000                  MOV     #STACK,SP       ;RESET STACK

012442  012737  000025  004172          MOV     #TTNO,@#TSTNM   ;THIS SAVES TEST NUMBER

012450  000167  000200                  JMP     TST26   ;JUMP TO NEXT TEST FOR RH70
012454  004767  025404                  JSR     PC,CLDISK       ;CLEAR REGISTERS LOAD R'S
012460  013746  001760                  MOV     @#UNIT,-(SP)
012464  052716  000100                  BIS     #IR,(SP)
012470  004737  037470                  JSR     PC,@#PUTREG     ;SAVE REGISTERS
012474  022637  001676                  CMP     (SP)+,@#CS2     ;IR SHOULD BE SET "OR" RESET
012500  001403                          BEQ     1$
012502  010237  001122                  MOV     R2,@#$BDADR     ;FAILING REGISTER RHCS2
012506  104011                          ERROR   11              ;RHCS2 DOES NOT HAVE IR
                                                                ;SET, UNIT NO. SET AND
                                                                ;ALL OTHER BITS 0
012510  005077  167104          1$:     CLR     @RHDB           ;LOAD DATA BUFFER (SILO) WITH 0
012514  012777  177777  167076          MOV     #-1,@RHDB       ;LOAD SILO WITH ALL ONES
012522  013737  001626  012532          MOV     @#RHCS2,@#2$    ;ADDRESS OF RHCS2
012530  104412                          WAT                     ;WAIT TRAP
012532  000000          2$:             .WORD                   ;ADDRESS OF RHCS2
012534  000200                          OR
012536  013746  001760          3$:     MOV     @#UNIT,-(SP)    ;
012542  052716  000300                  BIS     #OR!IR,(SP)     ;IR AND "OR"
012546  004737  037470                  JSR     PC,@#PUTREG     ;SAVE REGISTERS
012552  022637  001676                  CMP     (SP)+,@#CS2     ;IR AND "OR" SHOULD BE SET
012556  001403                          BEQ     4$
012560  010237  001122                  MOV     R2,@#$BDADR     ;SAVE RHCS2 ADDR. FAILING REG.
012564  104011                          ERROR   11              ;"OR" IN RHCS2 SHOULD BE
                                                                ;SET TOGETHER WITH IR AND
                                                                ;UNIT NO.
012566  017700  167026          4$:     MOV     @RHDB,R0        ;SAVE SILO DATA SHOULD BE 0
012572  017705  167022                  MOV     @RHDB,R5        ;SAVE SILO DATA SHOULD BE ALL 1
012576  022700  000000                  CMP     #0,R0           ;FIRST WORD 0? XYZ DO MORE TEST
012602  001410                          BEQ     5$              ;BRANCH IF YES
012604  005037  001124                  CLR     @#$GDDAT        ;GOOD DATA

012610  010037  001126                  MOV     R0,@#$BDDAT     ;BAD DATA
012614  013737  001620  037534          MOV     @#RHDB,@#REGADR ;SAVE RHDB FAILING REG.
012622  104001                          ERROR   1               ;SILO DID NOT HAVE THE FIRST WORD
                                                                ;"0" WHEN "OR" WAS SET
012624  022705  177777          5$:     CMP     #-1,R5          ;SECOND WORD ALL ONES?

012630  001411                          BEQ     TST26   ;BRANCH IF YES

012632  012737  177777  001124          MOV     #-1,@#$GDDAT    ;GOOD DATA
012640  010537  001126                  MOV     R5,@#$BDDAT     ;BAD DATA
```

```
4058  012644  013737  001620  037534          MOV     @#RHDB,@#REGADR ;SAVE RHDB FAILING REG.
4059  012652  104001                          ERROR   1               ;SILO DID NOT HAVE THE SECOND
4060                                                                  ;WORD OF ALL ONES WHEN "OR"
4061                                                                  ;WAS SET
4062
4063                                  ;;**************************************************************
4064                                  ;*TEST 26        SILO TEST 3
4065
4066                                  ;*      THIS TESTS SILO BUFFER BY FILLING IT WITH A COUNT FROM
4067                                  ;*      0 TO 65 AND THEN CHECKING IF IR IS DOWN AND "OR"
4068                                  ;*      IS HIGH AND COMPARING THE SILO OUTPUT.
4069
4070                                  ;;**************************************************************
4071  012654  000004                  TST26:  SCOPE
4072
4073  012656  012737  000026  004172          MOV     #TTNO,@#TSTNM   ;THIS SAVES TEST NUMBER
4074
4075  012664  000167  000216                  JMP     TST27   ;JUMP TO NEXT TEST FOR RH70
4076  012670  012700  046330                  MOV     #SILOTB,R0      ;TABLE POINTER
4077  012674  012705  000103                  MOV     #67.,R5         ;COUNTER
4078  012700  005020                  1$:     CLR     (R0)+           ;CLEAR TOTAL TABLE
4079  012702  005305                          DEC     R5              ;COUNT
4080  012704  001375                          BNE     1$              ;BRANCH IF NOT COMPLETELY CLEAR
4081  012706  004767  025152                  JSR     PC,CLDISK                       ;CLEAR ALL REG.
4082  012712  005000                          CLR     R0
4083  012714  012705  000102                  MOV     #66.,R5         ;COUNT
4084  012720  010077  166674          2$:     MOV     R0,@RHDB        ;LOAD SILO WITH COUNT FROM 0 TO 65
4085  012724  005200                          INC     R0              ;NEXT COUNT
4086  012726  005305                          DEC     R5              ;IS 66 LOADS DONE?
4087  012730  001373                          BNE     2$              ;BRANCH IF NOT.
4088  012732  013746  001760                  MOV     @#UNIT,-(SP)
4089  012736  052716  000200                  BIS     #OR,(SP)
4090  012742  004737  037470                  JSR     PC,@#PUTREG     ;SAVE REGISTERS
4091  012746  022637  001676                  CMP     (SP)+,@#CS2     ;"OR" SHOULD BE SET IR RESET
4092  012752  001405                          BEQ     3$              ;BRANCH IF YES
4093  012754  010237  001122                  MOV     R2,@#$BDADR     ;SAVE RHCS2 ADR. FAILING REG.
4094  012760  104011                          ERROR   11              ;"OR" WAS NOT SET, IR WAS NOT
4095  012762  005037  001772                  CLR     @#ERFLG$                ;RESET AFTER SILO WAS FULL
4096  012766  012700  046330          3$:     MOV     #SILOTB,R0      ;POINTER
4097  012772  012705  000102                  MOV     #66.,R5         ;COUNTER
4098  012776  017720  166616          4$:     MOV     @RHDB,(R0)+     ;READ SILO
4099  013002  005305                          DEC     R5              ;COUNT
4100  013004  001374                          BNE     4$              ;BRANCH IF 66 NOT DONE
4101  013006  012700  046330                  MOV     #SILOTB,R0      ;POINTER
4102  013012  012705  000102                  MOV     #66.,R5
4103  013016  005046                          CLR     -(SP)
4104  013020  021620                  5$:     CMP     (SP),(R0)+
4105  013022  001425                          BEQ     7$              ;BRANCH IF GOOD
4106  013024  014037  001126                  MOV     -(R0),@#$BDDAT  ;BAD DATA
4107  013030  011637  001124                  MOV     (SP),@#$GDDAT   ;GOOD DATA
4108  013034  013737  001620  037534          MOV     @#RHDB,@#REGADR ;FAILING REG. RHDB
4109  013042  005737  001772                  TST     @#ERFLG$                ;IS THIS FIRST ERROR?
4110  013046  001002                          BNE     6$              ;IF NOT BRANCH
4111  013050  104012                          ERROR   12              ;THESE TWO ERROR CALLS ARE FOR
4112  013052  000401                          BR      64$             ;BRANCH TO AVOID PRINTING NEXT ERROR
4113  013054  104013                  6$:     ERROR   13              ;THE SAME TYPEOUT.  SILO
```

```
                                                        ;HAD A COUNT WRITTEN IN.
                                                        ;ON READ OUT AN ERROR WAS
                                                        ;DETECTED. THE TOTAL SILO
                                                        ;READOUT IS IN LOCATION
                                                        ;"SILOTB" TO THE NEXT 65
                                                        ;WORDS.
013056  005720          64$:    TST     (R0)+           ;INCREMENT (R0)
                                                        ;ARE FURTHER COMPARES TO
013060  017746  166054          MOV     @SWR,-(SP)      ;BE DONE
013064  042716  177577          BIC     #^CSW07!SW08,(SP) ;ONLY KEEP SW7 AND SW8
013070  022726  000200          CMP     #SW07,(SP)+     ;TEST SW07
013074  001403                  BEQ     10$             ;IF NO MORE COMPARE THEN BRANCH
013076  005216          7$:     INC     (SP)            ;NEXT GOOD WORD
013100  005305                  DEC     R5              ;COUNT
013102  001346                  BNE     5$              ;BRANCH IF 66 NOT COMPLETE
013104  005726          10$:    TST     (SP)+           ;POP STACK

                        ;;**************************************************************
                        ;*TEST 27        SILO TEST4
                        ;*      NOW PUT 67 WORDS INTO SILO AND CHECK FOR DLT
                        ;*      EVEN AFTER THE 67TH. WORD INPUT THE FIRST WORD SHOULD NOT CHANGE

                        ;;**************************************************************
013106  000004          TST27:  SCOPE

013110  012737  000027  004172          MOV     #TTNO,@#TSTNM   ;THIS SAVES TEST NUMBER

013116  000167  000102          JMP     TST30   ;JUMP TO NEXT TEST FOR RH70
013122  004767  024736          JSR     PC,CLDISK       ;CLEAR DISK REG.
013126  005000                  CLR     R0              ;CLEAR R0
013130  005200          1$:     INC     R0              ;ADD 1
013132  010077  166462          MOV     R0,@RHDB        ;LOAD SILO
013136  022700  000103          CMP     #67.,R0         ;67 DONE?
013142  001401                  BEQ     2$              ;BRANCH IF YES
013144  000771                  BR      1$              ;NO SO BRANCH
013146  004737  037470  2$:     JSR     PC,@#PUTREG     ;SAVE REGISTERS

013152  032737  100000  001676          BIT     #DLT,@#CS2      ;DLT SET?
013160  001003                  BNE     3$              ;BRANCH IF YES
013162  010237  001122          MOV     R2,@#$BDADR     ;FAILING ADDRESS RHCS2
013166  104011                  ERROR   11              ;DATA LATE DID NOT SET AT 67TH.
013170  017737  166424  001126  3$:     MOV     @RHDB,@#$BDDAT  ;INPUT TO SILO
013176  012737  000001  001124          MOV     #1,@#$GDDAT     ;GOOD DATA
013204  023737  001124  001126          CMP     @#$GDDAT,@#$BDDAT ;COMPARE

013212  001404                  BEQ     TST30   ;BRANCH IF GOOD

013214  013737  001620  037534          MOV     @#RHDB,@#REGADR ;FAILING REG. RHDB
013222  104012                  ERROR   12              ;WORD IN RHDB CHANGED
                                                        ;AFTER THE 67TH INPUT.


                        ;;**************************************************************
                        ;*TEST 30        SILO TEST 5
                        ;*      THE SILO IS LOADED WITH 0,1,2,3 THEN AFTER
```

```
                                        ;*      'OR' IS UP A CLR IN RHCS2 IS DONE THEN 4,
                                        ;*      IS LOADED. AFTER 'OR' IS UP 2 READS FROM
                                        ;*      SILO IS DONE ON THE LAST DTL IN RHCS2 SHOULD BE SET
                                        ;;**********************************************************************
013224  000004                  TST30:  SCOPE
013226  012737  000030  004172          MOV     #TTNO,@#TSTNM   ;THIS SAVES TEST NUMBER
013234  000167  000256                  JMP     TST31   ;JUMP TO NEXT TEST FOR RH70
013240  004737  040064                  JSR     PC,@#CLDISK     ;CLEAR DISK
013244  013746  001760                  MOV     @#UNIT,-(SP)    ;GET UNIT NO.
013250  052716  000100                  BIS     #IR,(SP)        ;
013254  004737  037470                  JSR     PC,@#PUTREG     ;SAVE REGISTERS
013260  022637  001676                  CMP     (SP)+,@#CS2     ;IR SHOULD BE SET "OR" CLEARED
013264  001403                          BEQ     1$              ;BRANCH IF GOOD
013266  010237  001122                  MOV     R2,@#$BDADR     ;FAILING REGISTER RHCS2
013272  104011                          ERROR   11              ;RHCS2 DOES NOT HAVE IR SET
                                                                ;AND ALL OTHER BITS 0
013274  013700  001620          1$:     MOV     @#RHDB,R0       ;R0 HAS RHDB ADDRESS
013300  005001                          CLR     R1              ;DATA
013302  010110                  2$:     MOV     R1,@R0          ;0, THEN 1 THEN 2 THEN 3
                                                                ;IN RHDB
013304  005201                          INC     R1              ;INCREMENT DATA
013306  022701  000004                  CMP     #4,R1           ;IS 4 DONE
013312  103373                          BHIS    2$              ;BRANCH IF NOT
013314  013737  001626  013324          MOV     @#RHCS2,@#3$    ;
013322  104412                          WAT                     ;WAIT FOR "OR"
013324  000000                  3$:     .WORD   0               ;RHCS2 ADDRESS
013326  000200                          OR                      ;WAIT ON OR.
013330  004737  040064                  JSR     PC,@#CLDISK     ;CLR IN RHCS2
013334  013746  001760                  MOV     @#UNIT,-(SP)    ;UNIT NO.
013340  052716  000100                  BIS     #IR,(SP)
013344  004737  037470                  JSR     PC,@#PUTREG     ;SAVE REGISTERS
013350  022637  001676                  CMP     (SP)+,@#CS2     ;IR SHOULD BE SET "0"=0
013354  001403                          BEQ     4$              ;BRANCH IF GOOD
013356  010237  001122                  MOV     R2,@#$BDADR     ;FAILING REGISTER RHCS2
013362  104011                          ERROR   11              ;RHCS2 DOES NOT HAVE IR SET
                                                                ;AND ALL OTHER BITS 0
013364  013700  001620          4$:     MOV     @#RHDB,R0       ;R0 HAS RHDB ADDRESS
013370  012710  000004                  MOV     #4,@R0          ;LOAD 4 IN SILO
013374  011201                          MOV     @R2,R1          ;SAVE RHCS2
013376  011005                          MOV     @R0,R5          ;READ THE 4 IN SILO
013400  011003                          MOV     @R0,R3          ;READ SILO TO GET DLT
013402  011204                          MOV     @R2,R4          ;SAVE RHCS2
013404  032701  000200                  BIT     #OR,R1          ;TEST FOR OR IN RHCS2
013410  001424                          BEQ     6$              ;IF OR IS NOT SET BRANCH
013412  022705  000004                  CMP     #4,R5           ;SILO 4 IS NOW COMPARED
013416  001410                          BEQ     5$
013420  010037  037534                  MOV     R0,@#REGADR     ;SILO ADDRESS
013424  012737  000004  001124          MOV     #4,@#$GDDAT     ;GOOD DATA
013432  010537  001126                  MOV     R5,@#$BDDAT     ;BAD DATA
013436  104001                          ERROR   1               ;SILO DID NOT CONTAIN WORD
                                                                ;PUT IN AFTER "OR" WAS UP
013440  005703                  5$:     TST     R3              ;IS IT ZERO BECAUSE SILO
```

```
4226
4227  013442  001407                        BEQ     6$                      ;IS DESTRUCTIVE READ
4228  013444  010037  037534                MOV     R0,@#REGADR             ;BRANCH IF GOOD
4229  013450  005037  001124                CLR     @#$GDDAT                ;SILO ADDRESS
4230  013454  010337  001126                MOV     R3,@#$BDDAT             ;GOOD DATA
4231  013460  104001                        ERROR   1                       ;BAD DATA
4232                                                                        ;SILO SHOULD BE ZERO
4233                                                                        ;AFTER THE ONE WORD PUT IN
4234                                                                        ;HAS BEEN TAKEN OUT AS
4235  013462  032704  100000          6$:   BIT     #DLT,R4                 ;SILO IS A DESTRUCTIVE READ
4236                                                                        ;
4237  013466  001013                        BNE     TST31   ;BRANCH IF DLT SET
4238
4239  013470  013746  001760                MOV     @#UNIT,-(SP)    ;GET UNIT NO
4240  013474  052716  100300                BIS     #DLT!OR!IR,(SP) ;
4241  013500  012637  001124                MOV     (SP)+,@#$GDDAT  ;GOOD DATA
4242  013504  010437  001126                MOV     R4,@#$BDDAT     ;BAD DATA
4243  013510  010237  037534                MOV     R2,@#REGADR     ;RHCS2 ADDRESS
4244  013514  104001                        ERROR   1               ;DATA LATE ERROR
4245
4246
4247
4248
4249                                  ;;*****************************************************************
4250                                  ;*TEST 31       TEST ODD BYTE INSTRUCTION ON RHCS1
4251
4252                                  ;*      RDY (BIT 07) AND DVA (BIT 11) SHOULD ALWAYS BE SET
4253
4254                                  ;;*****************************************************************
4255  013516  000004                  TST31:  SCOPE
4256
4257
4258
4259
4260  013520  012737  000031  004172          MOV     #TTNO,@#TSTNM   ;THIS SAVES TEST NUMBER
4261
4262  013526  012706  001000                MOV     #STACK,SP       ;RESET STACK
4263  013532  004767  024326                JSR     PC,CLDISK                       ;CLEAR DISK REG.
4264  013536  012711  003566                MOV     #3566,  @R1     ;LOAD RHCS1 WITH ANY NUMBER
4265  013542  010146                        MOV     R1,     -(SP)   ;GETTING READY TO FORM ODD BYTE
4266  013544  005216                        INC     (SP)            ;SP NOW HAS ODD BYTE FOR RHCS1
4267  013546  112736  000005                MOVB    #5,     @(SP)+  ;MOVE 5 INTO ODD BYTE FOR RHCS1
4268  013552  011137  001126                MOV     @R1,    @#$BDDAT        ;TEST DATA
4269  013556  022737  004766  001126        CMP     #566!DVA!RDY,@#$BDDAT ;RHCS1 SHOULD HAVE 4766
4270  013564  001406                        BEQ     1$              ;BRANCH IF GOOD
4271  013566  012737  004766  001124        MOV     #566!DVA!RDY,@#$GDDAT ;GOOD DATA
4272  013574  010137  037534                MOV     R1,     @#REGADR ;FAILING REGISTER RHCS1
4273  013600  104001                        ERROR   1               ;MOVING A NUMBER INTO
4274                                                                ;ODD BYTE OF RHCS1 GAVE
4275                                                                ;WRONG RESULTS
4276  013602  112711  000032          1$:   MOVB    #32,    @R1     ;MOVE INTO EVEN BYTE
4277  013606  011137  001126                MOV     @R1,    @#$BDDAT        ;TEST DATA
4278  013612  022737  004632  001126        CMP     #432!DVA!RDY,@#$BDDAT ;RHCS1 SHOULD HAVE 4632
4279
4280  013620  001406                        BEQ     TST32   ;;BRANCH IF GOOD
4281
```

```
 4282  013622  012737  006632  001124          MOV     #2432!DVA!RDY,@#$GDDAT ;GOOD DATA
 4283  013630  010137  037534                  MOV     R1,     @#REGADR ;FAILING REGISTER RHCS1
 4284  013634  104001                          ERROR   1                ;MOVING A NUMBER INTO EVEN
 4285                                                                   ;BYTE OF RHCS1 GAVE WRONG
 4286                                                                   ;RESULTS
 4287
 4288
 4289
 4290                                  ;;*****************************************************************
 4291                                  ;*TEST 32        TEST ODD BYTE INSTRUCTION ON RHCS2
 4292
 4293                                  ;*      IR (BIT 06) AND THE UNIT SELECT (BIT 0-2) WILL BE SET
 4294
 4295                                  ;;*****************************************************************
 4296  013636  000004                  TST32:  SCOPE
 4297
 4298  013640  012737  000032  004172          MOV     #TTNO,@#TSTNM   ;THIS SAVES TEST NUMBER
 4299
 4300  013646  004767  024212                  JSR     PC,CLDISK
 4301  013652  052712  177000                  BIS     #177000,(R2)    ;LOAD RHCS2
 4302  013656  010246                          MOV     R2,     -(SP)   ;GETTING READY FOR ODD BYTE
 4303  013660  005216                          INC     (SP)            ;SP NOW HAS ODD BYTE FOR RHCS2
 4304  013662  105036                          CLRB    @(SP)+          ;CLERR RHCS2 ODD BYTE
 4305  013664  013746  001760                  MOV     @#UNIT, -(SP)   ;GET UNIT NO.
 4306  013670  052716  000100                  BIS     #IR,    (SP)    ;INPUT READY AS IT IS SET
 4307  013674  011237  001126                  MOV     @R2     ,@#$BDDAT       ;TEST DATA
 4308  013700  022637  001126                  CMP     (SP)+,  @#$BDDAT        ;COMPARE TO SEE THAT
 4309                                                                   ;"CLRB" DID CLEAR
 4310  013704  001411                          BEQ     1$
 4311  013706  013737  001760  001124          MOV     @#UNIT, @#$GDDAT
 4312  013714  052737  000100  001124          BIS     #IR,    @#$GDDAT ;GOOD DATA
 4313  013722  010237  037534                  MOV     R2,     @#REGADR ;FAILING REGISTER RHCS2
 4314  013726  104001                          ERROR   1                ;CLEARING ODD BYTE OF RHCS2
 4315                                                                   ;GAVE WRONG RESULTS
 4316  013730  013746  001760          1$:     MOV     @#UNIT, -(SP)
 4317  013734  052716  000010                  BIS     #BAI,   (SP)
 4318  013740  052712  020000                  BIS     #UPE,@R2        ;HAVE UPE AND MPE IN RHCS2
 4319                                                                   ;BESIDES UNIT SELECT
 4320  013744  112612                          MOVB    (SP)+,  @R2     ;MOVE INTO EVEN BYTE OF RHCS2
 4321  013746  013746  001760                  MOV     @#UNIT, -(SP)
 4322  013752  052716  000110                  BIS     #IR!BAI,(SP)
 4323  013756  011637  001124                  MOV     (SP),   @#$GDDAT ;GOOD DATA
 4324  013762  011237  001126                  MOV     @R2     ,@#$BDDAT       ;TEST DATA
 4325  013766  022637  001126                  CMP     (SP)+,  @#$BDDAT        ;COMPARE TO SEE THAT MOVB DID
 4326                                                                   ;MOVE EVEN BYTE ONLY
 4327
 4328  013772  001403                          BEQ     TST33   ;;BRANCH IF GOOD
 4329
 4330  013774  010237  037534                  MOV     R2,     @#REGADR ;FAILING REGISTER RHCS2
 4331  014000  104001                          ERROR   1                ;MOVING A NUMBER INTO EVEN
 4332                                                                   ;BYTE OF RHCS2 GAVE WRONG
 4333                                                                   ;RESULTS
 4334
 4335
 4336
 4337                                  ;;*****************************************************************
```

```
                                        ;*TEST 33       ODD BYTE TEST ON RHWC
                                        ;*
                                        ;*      IN THIS REGISTER NO BITS SHOULD BE PERMANENTLY SET
                                        ;
                                        ;
                                        ;;*****************************************************************
014002  000004                          TST33:  SCOPE
                                        ;
014004  012737  000033  004172          MOV     #TTNO,@#TSTNM   ;THIS SAVES TEST NUMBER
                                        ;
014012  012706  001000                  MOV     #STACK,SP       ;RESET STACK
014016  004767  024042                  JSR     PC,CLDISK                       ;CLEAR DISK REGSTERS
014022  013704  001622                  MOV     @#RHWC, R4      ;R4 NOW IS WORD COUNT REGISTER
014026  012714  025252                  MOV     #25252, @R4     ;LOAD RHWC
014032  010446                          MOV     R4,     -(SP)   ;GETTING READY TO FORM ODD BYTE
014034  005216                          INC     (SP)            ;SP NOW HAS ODD BYTE FOR RHWC
014036  112736  000377                  MOVB    #377,   @(SP)+  ;MOVE 377 INTO ODD BYTE OF RHWC
014042  011437  001126                  MOV     @R4,@#$BDDAT    ;TEST DATA
014046  022737  177652  001126          CMP     #177652,@#$BDDAT;COMPARE TO SEE IF MOVB DID OK
014054  001406                          BEQ     1$              ;BRANCH IF GOOD
014056  012737  177652  001124          MOV     #177652,@#$GDDAT        ;GOOD DATA
014064  010437  037534                  MOV     R4,     @#REGADR ;REGISTER FAILING RHWC
014070  104001                          ERROR   1               ;MOVING INTO ODD BYTE OF RHWC
                                                                ;GAVE WRONG RESULTS
014072  112714  000123          1$:     MOVB    #123,   @R4     ;MOVE INTO EVEN BYTE OF RHWC
014076  011437  001126                  MOV     @R4,@#$BDDAT    ;TEST DATA
014102  022737  177523  001126          CMP     #177523,@#$BDDAT
                                        ;
014110  001406                          BEQ     TST34   ;;BRANCH IF GOOD
                                        ;
014112  012737  177523  001124          MOV     #177523,@#$GDDAT        ;GOOD DATA
014120  010437  037534                  MOV     R4,     @#REGADR ;REGISTER FAILING RHWC
014124  104001                          ERROR   1
                                        ;
                                        ;
                                        ;*****************************************************************
                                        ;*TEST 34       TEST ODD BYTE INSTRUCTION ON RHBA
                                        ;
                                        ;*      BIT 0 SHOULD ALWAYS BE 0
                                        ;
                                        ;;*****************************************************************
014126  000004                          TST34:  SCOPE
014130  012706  001000                  MOV     #STACK,SP       ;RESET STACK
                                        ;
014134  012737  000034  004172          MOV     #TTNO,@#TSTNM   ;THIS SAVES TEST NUMBER
                                        ;
014142  004767  023716                  JSR     PC,CLDISK
014146  013704  001624                  MOV     @#RHBA, R4      ;R4 HAS ADDRESS OF RHBA
014152  012714  025253                  MOV     #25253, @R4     ;LOAD RHBA
014156  010446                          MOV     R4,     -(SP)   ;GETTING READY FOR ODD BYTE
014160  005216                          INC     (SP)            ;SP HAS ODD BYTE ADR. OF RHBA
014162  112736  000377                  MOVB    #377,   @(SP)+  ;LOAD ODD BYTE OF RHBA
014166  011437  001126                  MOV     @R4,@#$BDDAT    ;TEST DATA
014172  022737  177652  001126          CMP     #177652,@#$BDDAT        ;COMPARE MOVB RESULTS
014200  001406                          BEQ     1$              ;BRANCH IF GOOD
```

```
 4394  014202  012737  177652  001124          MOV     #177652,@#$GDDAT        ;GOOD DATA
 4395  014210  010437  037534                  MOV     R4,     @#REGADR ;FAILING REGISTER RHBA
 4396  014214  104001                          ERROR   1                ;MOVING INTO ODD BYTE OF
 4397                                                                   ;RHBA GAVE WRONG RESULTS
 4398  014216  112714  000125          1$:     MOVB    #125,   @R4
 4399  014222  011437  001126                  MOV     @R4,@#$BDDAT    ;TEST DATA
 4400  014226  022737  177524  001126          CMP     #177524,@#$BDDAT
 4401
 4402  014234  001406                          BEQ     TST35   ;;BRANCH IF GOOD
 4403
 4404  014236  012737  177524  001124          MOV     #177524,@#$GDDAT        ;GOOD DATA
 4405  014244  010437  037534                  MOV     R4,     @#REGADR ;FAILING REGISTER RHBA
 4406  014250  104001                          ERROR   1                ;MOVING INTO EVEN BYTE OF
 4407                                                                   ;RHBA GAVE WRONG RESULTS
 4408
 4409
 4410
 4411                                  ;;*****************************************************************
 4412                                  ;*      FOUR GENERAL REGISTERS WILL BE RESERVED FOR HARDWARE
 4413                                  ;*      R1=RHCS1 CONTROL AND STATUS1
 4414                                  ;*      R2=RHCS2 CONTROL AND STATUS2
 4415                                  ;*      R3=RHDS1 DRIVE STATUS 1
 4416                                  ;*      R4=RHER1 ERROR REGISTER1
 4417
 4418                                  ;*      WHENEVER ANY OTHER USE IS MADE OF THESE REGISTERS
 4419                                  ;*      APPROPIATE SAVING MUST BE DONE
 4420
 4421                                  ;;*****************************************************************
 4422
 4423
 4424
 4425
 4426
 4427
 4428
 4429                                  ;;*****************************************************************
 4430                                  ;*TEST 35       PACK ACKNOWLEDGE COMMAND TEST
 4431
 4432                                  ;*      THE PACK ACKNOWLEDGE COMMAND WILL BE LOADED INTO RHCS1 WITH GO
 4433                                  ;*      THEN ALL REGISTERS WILL BE CHECKED
 4434                                  ;*      RH CLEAR WILL BE GIVEN
 4435                                  ;*      THEN ALL REGISTERS WILL BE CHECKED
 4436
 4437                                  ;;*****************************************************************
 4438  014252  000004                  TST35:  SCOPE
 4439  014254  012706  001000                  MOV     #STACK,SP       ;RESET STACK
 4440
 4441  014260  012737  000035  004172          MOV     #TTNO,@#TSTNM   ;THIS SAVES TEST NUMBER
 4442
 4443
 4444  014266  004737  040064                  JSR     PC,@#CLDISK     ;INIT AND SET UP GENERAL REG.
 4445                                                                   ;AND UNIT NUMBER
 4446  014272  012777  000001  165350          MOV     #DMD,@RHMR      ;SET DIAGNOSTIC MODE
 4447
 4448  014300  013777  002054  165322          MOV     @#PKACK,@RHCS1  ;LOAD PACK ACKNOWLEDGE COMMAND INTO RHCS1
 4449
```

```
                                        ;SAVE REGISTERS FOR COMPARISON AFTER GO
014306  004037  040542                  JSR     R0,@#SAVER      ;SAVE
014312  001622                          RHWC                    ;FROM
014314  003126                          REINTO                  ;TO
014316  000023                          19.                     ;NUMBER OF REGISTERS SAVED

                                        ;GIVE GO TO PACK ACKNOWLEDGE COMMAND
014320  052777  000001  165302          BIS     #GO,@RHCS1      ;GO TO PACK ACKNOWLEDGE COMMAND

                                        ;CHANGE SAVED REGISTERS TO EXPECTED VALUES
014326  052737  000100  003156          BIS     #VV,@#REINTO+30 ;SAVED RHDS1

                                        ;AFTER GO HAS BEEN GIVEN TO PACK ACKNOWLEDGE COMMAND
                                        ;SAVE REGISTERS AGAIN SO THAT COMPARISONS CAN
                                        ;BE DONE
014334  004037  040542                  JSR     R0,@#SAVER      ;SAVE
014340  001622                          RHWC                    ;FROM
014342  002062                          WRFROM
014344  000023                          19.                     ;NUMBER OF REGISTERS SAVED

                                        ;AS UPPER BYTE OF RHAS CAN BE CHANGING IN A DUAL PORT
                                        ;OPERATION THE UPPER BYTE OF RHAS WILL BE SAVED AS IS
                                        ;SO THAT THE COMPARES ARE ONLY VALID FOR THE LOWER BYTE
014346  113737  003153  002107          MOVB    @#REINTO+25,@#WRFROM+25;SAVE UPPER RHAS


                                        ;COMPARE REGISTERS BEFORE PACK ACKNOWLEDGE COMMAND
                                        ;WITH AFTER GO
014354  004037  040736                  JSR     R0,@#COMPAR     ;COMPARE
014360  003126                          REINTO                  ;GOOD BUFFER
014362  002062                          WRFROM                  ;TEST BUFFER
014364  000023                          19.                     ;NUMBER
014366  014374                          1$                      ;RETURN FOR ERROR
014370  014374                          1$                      ;SAME
014372  014414                          2$                      ;RETURN FOR GOOD COMPARISON
014374  013705  044532          1$:     MOV     @#ERWORD,R5     ;GETTING READY TO INDEX
014400  060505                          ADD     R5,R5           ;DOUBLE ERROR WORD
014402  016537  001620  037534          MOV     RHWC-2(R5),@#REGADR ;FAILING REGISTER ADDRESS

014410  104001                          ERROR   1               ;IMPROPER REGISTER CHANGE
                                                                ;AFTER PACK ACKNOWLEDGE COMMAND
                                                                ;WITH GO IS GIVEN
014412  000207                          RTS     PC              ;RETURN TO COMPARISION

014414                          2$:


;;*********************************************************************
;*TEST 36        UNIBUS INIT TEST

;*      ALL POSSIBLE REGISTERS ARE FILLED WITH ONES
```

```
                                        ;*      A RESET COMMAND IS GIVEN
                                        ;*      ALL REGISTERS ARE CHECKED
                                        ;
                                        ;;*****************************************************************
014414  000004                        TST36:  SCOPE
014416  012706  001000                        MOV     #STACK,SP       ;RESET STACK
                                        ;
014422  012737  000036  004172                MOV     #TTNO,@#TSTNM   ;THIS SAVES TEST NUMBER
                                        ;
                                        ;
014430  004737  040064                        JSR     PC,@#CLDISK     ;INIT AND SET GENERAL REGISTERS
                                        ;
                                        ;FILL ALL POSSIBLE BITS WITH ONES
014434  012777  177777  165160                MOV     #177777,@RHWC   ;WORD COUNT REGISTER GETS 177777
014442  012777  177777  165154                MOV     #177777,@RHBA   ;BUS ADDRESS REGISTER GETS 177777
014450  052777  157010  165150                BIS     #157010,@RHCS2  ;CONTROL AND STATUS 2 GETS 177430
014456  012777  001476  165144                MOV     #1476,@RHCS1    ;CONTROL AND STATUS REGISTER/GETS 21476
014464  012777  177777  165140                MOV     #177777,@RHER1  ;ERROR REGISTER1 GETS 177777
014472  012777  017437  165134                MOV     #17437,@RHDST   ;DESIRED SECTOR TRACK
014500  012777  177777  165130                MOV     #177777,@RHER2  ;ERROR REGISTER 2
014506  012777  016277  165124                MOV     #16277,@RHOF    ;OFFSET REGISTER
014514  012777  000777  165120                MOV     #777,@RHCA      ;DESIRED CYLINDER
014522  012777  177777  165114                MOV     #177777,@RHER3  ;ERROR REGISTER 3
014530  012777  000001  165112                MOV     #DMD,@RHMR      ;MAINTENANCE REGISTER
014536  012777  177777  165104                MOV     #177777,@RHMR   ;MAINTENANCE REGISTER
                                        ;
                                        ;NOW SAVE REGISTERS IN READ INTO BUFFER
014544  004037  040542                        JSR     R0,@#SAVER      ;SAVE
014550  001622                                RHWC                    ;FROM
014552  003126                                REINTO                  ;TO
014554  000021                                17.                     ;NUMBER
                                        ;
                                        ;GIVE RESET AND REINSTATE UNIT NUMBER
014556  000005                                RESET
014560  004737  052030                        JSR     PC,@#$TKINT     ;INITILIZE TK
014564  053777  001760  165034                BIS     @#UNIT,@RHCS2
                                        ;
                                        ;CHANGE SAVED REGISTERS TO EXPECTED VALUE
014572  005037  003130                        CLR     @#REINTO+2      ;CLEAR SAVED RHBA
014576  013746  001760                        MOV     @#UNIT,-(SP)    ;GET UNIT NUMBER FRO SAVED RHCS2
014602  052716  000100                        BIS     #IR,(SP)        ;INCLUDE IR
014606  012637  003132                        MOV     (SP)+,@#REINTO+4 ;SAVED RHCS2
014612  012737  004276  003134                MOV     #DVA!RDY!76,@#REINTO+6 ;SAVED RHCS1
014620  005037  003136                        CLR     @#REINTO+10     ;SAVED RHER1
014624  005037  003142                        CLR     @#REINTO+14     ;SAVED RHER2
014630  012737  116000  003144                MOV     #116000,@#REINTO+16;SAVED RHOF
014636  005037  003150                        CLR     @#REINTO+22     ;SAVED RHER3
014642  105037  003152                        CLRB    @#REINTO+24     ;SAVED RHAS
014646  012737  000400  003154                MOV     #400,@#REINTO+26;SAVED RHMR
                                        ;CHANGE RHDS1 WITHOUT CHANGING PROG BIT
014654  013746  003156                        MOV     @#REINTO+30,-(SP)       ;GET RHDS1
014660  042716  176777                        BIC     #^CPROG,(SP)            ;CLEAR EVERYTHING EXCEPT PROG
014664  052716  000700                        BIS     #700,(SP)               ;SET EXPECTED BITS
                                        ;
                                        ;
014670  012637  003156                        MOV     (SP)+,@#REINTO+30;SAVED RHDS1
```

```
4562  014674  005037  003164                  CLR     @#REINTO+36     ;SAVED RHEC1
4563  014700  005037  003166                  CLR     @#REINTO+40     ;SAVED RHEC2
4564
4565                                          ;AFTER RESET SAVE REGISTERS FOR COMPARISONS TO BE DONE
4566  014704  004037  040542                  JSR     R0,@#SAVER      ;SAVE
4567  014710  001622                          RHWC                    ;FROM
4568  014712  002062                          WRFROM                  ;TO
4569  014714  000021                          17.                     ;NUMBER
4570
4571                                          ;AS UPPER BYTE OF RHAS CAN BE CHANGING IN A DUAL PORT
4572                                          ;OPERATION THE UPPER BYTE OF RHAS WILL BE SAVED AS IS
4573                                          ;SO THAT THE COMPARES ARE ONLY VALID FOR THE LOWER BYTE
4574  014716  113737  003153  002107          MOVB    @#REINTO+25,@#WRFROM+25;SAVE UPPER RHAS .
4575
4576
4577                                          ;AS UPPER BYTE OF RHAS CAN BE CHANGING IN A DUAL PORT
4578                                          ;OPERATION THE UPPER BYTE OF RHAS WILL BE SAVED AS IS
4579                                          ;SO THAT THE COMPARES ARE ONLY VALID FOR THE LOWER BYTE
4580  014724  113737  003153  002107          MOVB    @#REINTO+25,@#WRFROM+25;SAVE UPPER RHAS
4581
4582
4583                                          ;COMPARE REGISTERS BEFORE RESET WITH AFTER
4584  014732  004037  040736                  JSR     R0,@#COMPAR     ;COMPARE
4585  014736  003126                          REINTO                  ;GOOD BUFFER
4586  014740  002062                          WRFROM                  ;TEST BUFFER
4587  014742  000021                          17.                     ;NUMBER
4588  014744  014752                          1$                      ;RETURN FOR ERROR
4589  014746  014752                          1$                      ;SAME
4590  014750  014772                          2$                      ;RETURN FOR GOOD COMPARISON
4591
4592  014752  013705  044532          1$:     MOV     @#ERWORD,R5     ;GETTING READY TO INDEX
4593  014756  060505                          ADD     R5,R5           ;DOUBLE ERROR WORD
4594  014760  016537  001620  037534          MOV     RHWC-2(R5),@#REGADR ;FAILING REGISTER ADDRESS
4595  014766  104001                          ERROR   1               ;REGISTER CONTENTS AFTER
4596                                                                  ;A RESET THAT IS AN
4597                                                                  ;UNIBUS INITALIZE CAUSED
4598                                                                  ;AN IMPROPER REGISTER CHANGE
4599  014770  000207                          RTS     PC              ;RETURN TO COMPARISON
4600  014772                          2$:                             ;RETURN POINT ON GOOD COMPARISON
4601
4602
4603
4604
4605                                          ;;*****************************************************************
4606                                          ;*TEST 37       READ IN PRESET
4607
4608                                          ;*      ALL POSSIBLE REGISTERS WILL BE FILLED WITH ONES
4609                                          ;*      THE REGISTER CONTENTS WILL BE SAVED IN REINTO BUFFER
4610                                          ;*      THE READ IN PRESET COMMAND WILL BE GIVEN
4611                                          ;*      ALL REGISTERS WILL BE CHECKED
4612
4613                                          ;;*****************************************************************
4614  014772  000004                  TST37:  SCOPE
4615  014774  012706  001000                  MOV     #STACK,SP       ;RESET STACK
4616
4617  015000  012737  000037  004172          MOV     #TTNO,@#TSTNM   ;THIS SAVES TEST NUMBER
```

```
                                ;FILL ALL POSSIBLE BITS WITH ONES

015012  012777  177777  164602   MOV     #177777,@RHWC   ;WORD COUNT REGISTER GETS 177777
015020  012777  177777  164576   MOV     #177777,@RHBA   ;BUS ADDRESS REGISTER GETS 177777
015026  012777  017437  164600   MOV     #17437,@RHDST   ;DESIRED SECTOR TRACK GETS 17437
015034  012777  016377  164576   MOV     #16377,@RHOF    ;OFFSET REGISTER GETS 16277
015042  012777  000777  164572   MOV     #777,@RHCA      ;DESIRED CYLINDER GETS 777
015050  012746  001400           MOV     #A16!A17,-(SP)  ;GET BIT 9 AND 8
015054  053716  002056           BIS     @#READIN,(SP)
015060  012677  164544           MOV     (SP)+,@RHCS1    ;FILL READ IN PRESET IN RHCS1
015064  012777  000001  164556   MOV     #DMD,@RHMR      ;SET DIAGNOSTIC MODE

                                ;THE REGISTERS WILL BE SAVED IN REINTO BUFFER
015072  004037  040542           JSR     R0,@#SAVER      ;SAVE
015076  001622                   RHWC                    ;FROM
015100  003126                   REINTO                  ;TO
015102  000021                   17.                     ;NUMBER SAVED

                                ;GIVE READ IN PRESET COMMAND
015104  052777  000001  164516   BIS     #GO,@RHCS1      ;INCLUDE GO TO READ IN PRESET

                                ;NOW SAVED REGISTERS WILL BE CHANGED TO EXPECTED VALUE
015112  005037  003140           CLR     @#REINTO+12     ;CLEAR SAVED RHDST
015116  042737  016000  003144   BIC     #FMT22!HCI!ECI,@#REINTO+16 ;CLEAR FMT22,HCI,ECI IN
                                                        ;SAVED RHOF
015124  052737  000100  003144   BIS     #VV,@#REINTO+16 ;SET VV IN SAVED RHOF
015132  005037  003146           CLR     @#REINTO+20     ;CLEAR SAVED RHCA

                                ;AFTER A READ IN PRESET COMMAND
                                ;SAVE REGISTERS AGAIN SO THAT COMPARES CAN BE DONE
015136  004037  040542           JSR     R0,@#SAVER      ;SAVE
015142  001622                   RHWC                    ;FROM
015144  002062                   WRFROM                  ;TO
015146  000021                   17.                     ;NUMBER OF REGISTERS SAVED

                                ;AS UPPER BYTE OF RHAS CAN BE CHANGING IN A DUAL PORT
                                ;OPERATION THE UPPER BYTE OF RHAS WILL BE SAVED AS IS
                                ;SO THAT THE COMPARES ARE ONLY VALID FOR THE LOWER BYTE
015150  113737  003153  002107   MOVB    @#REINTO+25,@#WRFROM+25;SAVE UPPER RHAS

                                ;COMPARE REGISTERS BEFORE READ IN PRESET COMMAND
                                ;WITH AFTER COMMAND
015156  004037  040736           JSR     R0,@#COMPAR     ;COMPARE
015162  003126                   REINTO                  ;GOOD BUFFER
015164  002062                   WRFROM                  ;TEST BUFFER
015166  000021                   17.                     ;NUMBER OF REGISTERS
015170  015176                   1$                      ;RETURN FOR ERROR
015172  015176                   1$                      ;SAME
015174  015216                   2$                      ;RETURN FOR GOOD COMPARISON

015176  013705  044532    1$:    MOV     @#ERWORD,R5     ;GETTING READY TO INDEX
```

```
 4674  015202  060505                        ADD     R5,R5               ;DOUBLE ERROR WORD
 4675  015204  016537  001620  037534        MOV     RHWC-2(R5),@#REGADR ;FAILING REG. ADDRESS
 4676  015212  104001                        ERROR   1                   ;READ IN PRESET CAUSED IMPROPER
 4677                                                                    ;REGISTER CHANGE
 4678  015214  000207                        RTS     PC                  ;RETURN FOR FURTHER COMPARISONS
 4679
 4680  015216                        2$:                                 ;NO ERRORS
 4681
 4682
 4683
 4684
 4685
 4686
 4687
 4688                                ;;*****************************************************************
 4689                                ;*TEST 40       NO OPERATION FUNCTION TEST
 4690                                ;*      ALL POSSIBLE REGISTERS ARE CLEARED THEN A"NOP"=0
 4691                                ;*      IS GIVEN NO CHANGE SHOULD HAPPEN
 4692                                ;*      ALL POSSIBLE REGISTERS ARE FILLED WITH ONES THEN A "NOP"
 4693                                ;*      IS GIVEN NO CHANGE SHOULD HAPPEN
 4694
 4695                                ;;*****************************************************************
 4696  015216  000004                TST40:  SCOPE
 4697
 4698  015220  012737  000040  004172        MOV     #TTNO,@#TSTNM   ;THIS SAVES TEST NUMBER
 4699
 4700                                        ;START WITH CLR IN RHCS2 (BIT5)
 4701  015226  004737  040064                JSR     PC,@#CLDISK     ;CLEAR ALL POSSIBLE BITS
 4702  015232  012777  000001  164410        MOV     #DMD,@RHMR      ;SET DIAGNOSTIC MODE
 4703  015240  013711  002016                MOV     @#NOPERA,@R1    ;PUT NOP OPERATION=0 IN RHCS1
 4704  015244  012700  001622                MOV     #RHWC,R0        ;STARTING ADDRESS OF REG
 4705  015250  012703  001672                MOV     #WC,R3          ;STARTING ADDRESS OF WHERE SAVED
 4706  015254  012702  000021                MOV     #RHEC2-RHWC+2/2,R2 ;NUMBER OF REGISTERS
 4707  015260  013023                1$:     MOV     @(R0)+,(R3)+    ;SAVE HARDWARE REG
 4708  015262  005302                        DEC     R2              ;COUNT
 4709  015264  001375                        BNE     1$              ;BRANCH IF NOT COMPLETE
 4710  015266  013737  001652  015306        MOV     @#RHDS1,@#2$    ;GET ADDRESS OF DRIVE STATUS
 4711  015274  010137  015314                MOV     R1,@#3$ ;GET ADDRESS OF RHCS1
 4712  015300  052711  000001                BIS     #GO,@R1         ;GO TO RHCS1
 4713  015304  104412                        WAT                     ;WAIT FOR DRY IN RHDS1
 4714  015306  000000                2$:     .WORD   0               ;ADDRESS OF DRIVE STATUS RHDS1
 4715  015310  000200                        DRY                     ;DRY WILL BE WAITED ON
 4716  015312  104412                        WAT                     ;WAIT FOR RDY IN RHCS1
 4717  015314  000000                3$:     .WORD   0               ;ADDRESS OF RHCS1 PUT HERE BY AN
 4718                                                                  ;EARLIER MOV
 4719  015316  000200                        RDY                     ;RDY WILL BE WAITED ON
 4720
 4721
 4722
 4723                                        ;AFTER A NO OP  COMMAND
 4724                                        ;SAVE REGISTERS AGAIN SO THAT COMPARES CAN BE DONE
 4725  015320  004037  040542                JSR     R0,@#SAVER      ;SAVE
 4726  015324  001622                        RHWC                    ;FROM
 4727  015326  002062                        WRFROM                  ;TO
 4728  015330  000021                        17.                     ;NUMBER OF REGISTERS SAVED
 4729
```

```
                                        ;AS UPPER BYTE OF RHAS CAN BE CHANGING IN A DUAL PORT
                                        ;OPERATION THE UPPER BYTE OF RHAS WILL BE SAVED AS IS
                                        ;SO THAT THE COMPARES ARE ONLY VALID FOR THE LOWER BYTE
015332  113737  001717  002107        MOVB    @#AS+1,@#WRFROM+25;SAVE UPPER RHAS


                                        ;COMPARE REGISTERS BEFORE  NO OP COMMAND
                                        ;WITH AFTER COMMAND
015340  004037  040736                JSR     R0,@#COMPAR     ;COMPARE
015344  001672                        WC                      ;GOOD BUFFER
015346  002062                        WRFROM                  ;TEST BUFFER
015350  000021                        17.                     ;NUMBER OF REGISTERS
015352  015360                        4$                      ;RETURN FOR ERROR
015354  015360                        4$                      ;SAME
015356  015400                        5$                      ;RETURN FOR GOOD COMPARISON

015360  013705  044532        4$:     MOV     @#ERWORD,R5     ;GETTING READY TO INDEX
015364  060505                        ADD     R5,R5           ;DOUBLE ERROR WORD
015366  016537  001620  037534        MOV     RHWC-2(R5),@#REGADR ;FAILING REG. ADDRESS
015374  104001                        ERROR   1               ;NO OP COMMAND CAUSED IMPROPER
                                                                ;REGISTER CHANGE
015376  000207                        RTS     PC              ;RETURN FOR FURTHER COMPARISONS

015400                          5$:                             ;NO ERRORS


015400  012737  015406  001110        MOV     #14$,@#$LPERR   ;SET SCOPE LOOP TO 14$
015406  004737  040064        14$:    JSR     PC,@#CLDISK     ;INIT LAST ALL ZERO TEST
015412  012777  000001  164230        MOV     #DMD,@RHMR      ;SET DIAGNOSTIC MODE


                                        ;NOW START WITH ALL ONES IN ALL POSSIBLE REGISTERS

015420  012700  001622                MOV     #RHWC,R0        ;ADDRESS OF FIRST REGISTER
015424  012705  000021                MOV     #RHEC2-RHWC+2/2,R5 ;NO. OF REGISTERS
015430  012730  177676        6$:     MOV     #177676,@(R0)+  ;FILL WITH ALL ONES
015434  013777  001760  164164        MOV     @#UNIT,@RHCS2   ;REINSTATE UNIT NUMBER UNDER TEST
                                                                ;KEEP INTERRUPT DISABLED
015442  005305                        DEC     R5              ;COUNT
015444  001371                        BNE     6$              ;BRANCH IF INCOMPLETE
015446  013711  002016                MOV     @#NOPERA,@R1    ;PUT NOP OPERATION =0 IN RHCS1
015452  012700  001622                MOV     #RHWC,R0        ;STARTING ADDRESS OF REG
015456  012703  001672                MOV     #WC,R3          ;STARTING ADDRESS OF WHERE SAVED
015462  012702  000021                MOV     #RHEC2-RHWC+2/2,R2 ;NUMBER OF REGISTERS
015466  013023                7$:     MOV     @(R0)+,(R3)+    ;SAVE HARDWARE REG
015470  005302                        DEC     R2              ;COUNT
015472  001375                        BNE     7$              ;BRANCH IF NOT COMPLETE
015474  013737  001652  015514        MOV     @#RHDS1,@#10$   ;GET ADDRESS OF DRIVE STATUS
015502  010137  015522                MOV     R1,@#11$        ;GET ADDRESS OF RHCS1
015506  052711  000001                BIS     #GO,@R1         ;GO TO RHCS1
015512  104412                        WAT                     ;WAIT FOR DRY IN RHDS1
015514  000000                10$:    .WORD   0               ;ADDRESS OF DRIVE STATUS RHDS1
015516  000200                        DRY                     ;DRY WILL BE WAITED ON
015520  104412                        WAT                     ;WAIT FOR RDY IN RHCS1
015522  000000                11$:    .WORD   0               ;ADDRESS OF RHCS1 PUT HERE BY AN
```

```
                                                        ;EARLIER MOV.
015524  000200                          RDY             ;RDY WILL BE WAITED ON


                                        ;AFTER A NO OP COMMAND
                                        ;SAVE REGISTERS AGAIN SO THAT COMPARES CAN BE DONE
015526  004037  040542                  JSR     R0,@#SAVER      ;SAVE
015532  001622                          RHWC                    ;FROM
015534  002062                          WRFROM                  ;TO
015536  000021                          17.                     ;NUMBER OF REGISTERS SAVED

                                        ;AS UPPER BYTE OF RHAS CAN BE CHANGING IN A DUAL PORT
                                        ;OPERATION THE UPPER BYTE OF RHAS WILL BE SAVED AS IS
                                        ;SO THAT THE COMPARES ARE ONLY VALID FOR THE LOWER BYTE
015540  113737  001717  002107          MOVB    @#AS+1,@#WRFROM+25;SAVE UPPER RHAS


                                        ;COMPARE REGISTERS BEFORE NO OP COMMAND
                                        ;WITH AFTER COMMAND
015546  004037  040736                  JSR     R0,@#COMPAR     ;COMPARE
015552  001672                          WC                      ;GOOD BUFFER
015554  002062                          WRFROM                  ;TEST BUFFER
015556  000021                          17.                     ;NUMBER OF REGISTERS
015560  015566                          12$                     ;RETURN FOR ERROR
015562  015566                          12$                     ;SAME
015564  015606                          13$                     ;RETURN FOR GOOD COMPARISON

015566  013705  044532          12$:    MOV     @#ERWORD,R5     ;GETTING READY TO INDEX
015572  060505                          ADD     R5,R5           ;DOUBLE ERROR WORD
015574  016537  001620  037534          MOV     RHWC-2(R5),@#REGADR ;FAILING REG. ADDRESS
015602  104001                          ERROR   1               ;NO OP COMMAND CAUSED IMPROPER
                                                                ;REGISTER CHANGE
015604  000207                          RTS     PC              ;RETURN FOR FURTHER COMPARISONS

015606                          13$:                            ;NO ERRORS


;;**********************************************************************
;*TEST 41        DRIVE CLEAR

;*      ALL WRITE BITS OF ALL REGISTERS EXCEPT RHDB ARE FILLED WITH
;*      ONES EXCEPT FOR BIT #0 AND BIT #6 WHICH ARE "GO" AND
;*      "ENABLE INTERRUPT" BITS
;*      THEN A DRIVE CLEAR IS PERFORMED
;*      THEN ALL REGISTERS EXCEPT RHDB ARE CHECKED

;;**********************************************************************
015606  000004                  TST41:  SCOPE
015610  012706  001000                  MOV     #STACK,SP       ;RESET STACK
```

```
4842  015614  012737  000041  004172          MOV     #TTNO,@#TSTNM   ;THIS SAVES TEST NUMBER
4843
4844  015622  004737  040064                  JSR     PC,@#CLDISK     ;SET REGISTERS AND CLEAR
4845                                          ;FILL ALL POSSIBLE BITS WITH ONES
4846  015626  012777  177777  163764          MOV     #177777,@RHDB   ;BUS ADDRESS REGISTER GETS 177777
4847  015634  012777  177777  163760          MOV     #177777,@RHWC   ;WORD COUNT REGISTER GETS 177777
4848  015642  012777  177777  163754          MOV     #177777,@RHBA   ;BUS ADDRESS REGISTER GETS 177777
4849  015650  052777  157010  163750          BIS     #157010,@RHCS2  ;CONTROL AND STATUS 2 GETS 157010
4850  015656  012777  001476  163744          MOV     #1476,@RHCS1    ;CONTROL AND STATUS REGISTER/GETS 1476
4851  015664  012777  177777  163740          MOV     #177777,@RHER1  ;ERROR REGISTER1 GETS 177777
4852  015672  012777  017437  163734          MOV     #17437,@RHDST   ;DESIRED SECTOR TRACK
4853  015700  012777  177777  163730          MOV     #177777,@RHER2  ;ERROR REGISTER 2
4854  015706  012777  016277  163724          MOV     #16277,@RHOF    ;OFFSET REGISTER
4855  015714  012777  177777  163720          MOV     #177777,@RHCA   ;DESIRED CYLINDER
4856  015722  012777  177777  163714          MOV     #177777,@RHER3  ;ERROR REGISTER 3
4857  015730  012777  000001  163712          MOV     #DMD,@RHMR      ;MAINTENANCE REGISTER
4858  015736  012777  177777  163704          MOV     #177777,@RHMR   ;MAINTENANCE REGISTER
4859
4860
4861
4862                                  ;THIS SETS BITS FOR ALL PRESENT DRIVES
4863
4864  015744  013700  002004                  MOV     @#TOTALAT,R0    ;GET DRIVE PRESENT
4865  015750  005012                          CLR     @R2             ;CLEAR RHCS2 AND CARRY BIT
4866  015752  012705  000010                  MOV     #8.,R5          ;COUNTER
4867  015756  006000          30$:    ROR     R0              ;GET BIT INTO CARRY
4868  015760  103002                  BCC     31$             ;BRANCH IF NO UNIT ON THIS BIT
4869  015762  012714  177777                  MOV     #-1,@R4 ;MOVE INTO ERROR REGISTER TO SET ATA
4870  015766  005212          31$:    INC     @R2             ;INCREMENT RHCS2 - UNIT NO.
4871  015770  005305                  DEC     R5              ;COUNT
4872  015772  001401                  BEQ     27$             ;BRANCH IF 8 DONE
4873  015774  000770                  BR      30$             ;CONTINUE THIS ROUTINE
4874  015776  013746  001760  27$:    MOV     @#UNIT,-(SP)
4875  016002  052716  157010          BIS     #157010,(SP)    ;REINSTATE SET BITS
4876  016006  012612                  MOV     (SP)+,@R2       ;
4877
4878
4879  016010  012777  000001  163632          MOV     #DMD,@RHMR      ;SET DMD
4880  016016  013711  002024                  MOV     @#DCLEAR,@R1    ;DRIVE CLEAR = 10 INTO RHCS1
4881  016022  052711  000001                  BIS     #GO,@R1         ;GO
4882  016026  012700  001620                  MOV     #RHDB,  R0      ;R0 CONTAINS ADDR. OF ADDR. OF REG.
4883
4884
4885
4886                                  ;DATA BUFFER REGISTER
4887
4888
4889  016032  012737  177777  001124  28$:    MOV     #177777,        @#$GDDAT ;GOOD DATA FOR ERROR TYPEOUT
4890  016040  013037  001126                  MOV     @(R0)+,@#$BDDAT ;TEST DATA
4891  016044  022737  177777  001126          CMP     #177777,        @#$BDDAT;COMPARE DATA
4892  016052  001402                  BEQ     3$              ;BRANCH IF GOOD
4893  016054  004737  016712                  JSR     PC,@#ERCLFC     ;JUMP TO ERROR FOR CLR (BIT 5)
4894                                                          ;IN RHCS2
4895
4896
4897                                  ;WORD COUNT REGISTER
```

```
 4898
 4899
 4900  016060  012737  177777  001124  3$:   MOV   #177777,   @#$GDDAT ;GOOD DATA FOR ERROR TYPEOUT
 4901  016066  013037  001126              MOV   @(R0)+,@#$BDDAT ;TEST DATA
 4902  016072  022737  177777  001126      CMP   #177777,   @#$BDDAT;COMPARE DATA
 4903  016100  001402                      BEQ   4$          ;BRANCH IF GOOD
 4904  016102  004737  016712              JSR   PC,@#ERCLFC ;JUMP TO ERROR FOR CLR (BIT 5)
 4905                                                        ;IN RHCS2
 4906
 4907
 4908                                      ;BUS ADDRESS REGISTER
 4909
 4910
 4911  016106  012737  177776  001124  4$:   MOV   #177776,   @#$GDDAT ;GOOD DATA FOR ERROR TYPEOUT
 4912  016114  013037  001126              MOV   @(R0)+,@#$BDDAT ;TEST DATA
 4913  016120  022737  177776  001126      CMP   #177776,   @#$BDDAT;COMPARE DATA
 4914  016126  001402                      BEQ   5$          ;BRANCH IF GOOD
 4915  016130  004737  016712              JSR   PC,@#ERCLFC ;JUMP TO ERROR FOR CLR (BIT 5)
 4916                                                        ;IN RHCS2
 4917
 4918
 4919                                      ;CONTROL AND STATUS 2 REGISTER
 4920
 4921
 4922  016134  012746  000110          5$:   MOV   #110,    -(SP)   ;INCLUDE IR
 4923  016140  053716  001760              BIS   @#UNIT, (SP)    ;SET UNIT NO.
 4924  016144  012637  001124              MOV   (SP)+,  @#$GDDAT;GOOD DATA FOR TYPE OUT
 4925  016150  013037  001126              MOV   @(R0)+,@#$BDDAT ;TEST DATA
 4926  016154  023737  001124  001126      CMP   @#$GDDAT,@#$BDDAT;COMPARE DATA
 4927  016162  001402                      BEQ   6$          ;BRANCH IF GOOD
 4928  016164  004737  016712              JSR   PC,@#ERCLFC ;JUMP TO ERROR FOR CLR (BIT 5)
 4929                                                        ;IN RHCS2
 4930
 4931
 4932
 4933
 4934
 4935
 4936                                      ;CONTROL AND STATUS 1 REGISTER
 4937
 4938  016170  005737  001764          6$:   TST   @#NUNIT         ;ARE THERE MORE THAN ONE UNIT
 4939  016174  001404                      BEQ   32$         ;BRANCH IF ONLY ONE UNIT
 4940  016176  012737  104210  001124      MOV   #104210,@#$GDDAT        ;GOOD DATA
 4941  016204  000403                      BR    33$
 4942  016206  012737  004210  001124  32$:   MOV   #4210,@#$GDDAT  ;GOOD DATA
 4943  016214  013037  001126          33$:   MOV   @(R0)+,@#$BDDAT ;TEST DATA
 4944
 4945  016220  023737  001124  001126      CMP   @#$GDDAT,@#$BDDAT;COMPARE DATA
 4946  016226  001402                      BEQ   7$          ;BRANCH IF GOOD
 4947  016230  004737  016712              JSR   PC,@#ERCLFC ;JUMP TO ERROR FOR CLR BIT 5
 4948                                                        ;IN RHCS2
 4949
 4950                                      ;ERROR 1 REGISTER
 4951
 4952
 4953  016234  012737  000000  001124  7$:   MOV   #0,     @#$GDDAT ;GOOD DATA FOR ERROR TYPEOUT
```

```
4954  016242  013037  001126                    MOV     @(R0)+,@#$BDDAT ;TEST DATA
4955  016246  022737  000000  001126            CMP     #0,     @#$BDDAT;COMPARE DATA
4956  016254  001402                            BEQ     10$             ;BRANCH IF GOOD
4957  016256  004737  016712                    JSR     PC,@#ERCLFC     ;JUMP TO ERROR FOR CLR (BIT 5)
4958                                                                    ;IN RHCS2
4959
4960                                            ;DESIRED SECTOR/TRACK REGISTER
4961
4962
4963  016262  012737  017437  001124  10$:     MOV     #17437, @#$GDDAT ;GOOD DATA FOR ERROR TYPEOUT
4964  016270  013037  001126                    MOV     @(R0)+,@#$BDDAT ;TEST DATA
4965  016274  022737  017437  001126            CMP     #17437, @#$BDDAT;COMPARE DATA
4966  016302  001402                            BEQ     11$             ;BRANCH IF GOOD
4967  016304  004737  016712                    JSR     PC,@#ERCLFC     ;JUMP TO ERROR FOR CLR (BIT 5)
4968                                                                    ;IN RHCS2
4969
4970                                            ;ERROR 2 REGISTER
4971
4972
4973  016310  012737  000000  001124  11$:     MOV     #0,     @#$GDDAT ;GOOD DATA FOR ERROR TYPEOUT
4974  016316  013037  001126                    MOV     @(R0)+,@#$BDDAT ;TEST DATA
4975  016322  022737  000000  001126            CMP     #0,     @#$BDDAT;COMPARE DATA
4976  016330  001402                            BEQ     12$             ;BRANCH IF GOOD
4977  016332  004737  016712                    JSR     PC,@#ERCLFC     ;JUMP TO ERROR FOR CLR (BIT 5)
4978                                                                    ;IN RHCS2
4979
4980                                            ;OFFSET REGISTER
4981
4982
4983  016336  012737  116000  001124  12$:     MOV     #116000,        @#$GDDAT ;GOOD DATA FOR ERROR TYPEOUT
4984  016344  013037  001126                    MOV     @(R0)+,@#$BDDAT ;TEST DATA
4985  016350  022737  116000  001126            CMP     #116000,        @#$BDDAT;COMPARE DATA
4986  016356  001402                            BEQ     13$             ;BRANCH IF GOOD
4987  016360  004737  016712                    JSR     PC,@#ERCLFC     ;JUMP TO ERROR FOR CLR (BIT 5)
4988                                                                    ;IN RHCS2
4989
4990                                            ;DESIRED CYLINDER ADDRESS REGISTER
4991
4992
4993  016364  012737  001777  001124  13$:     MOV     #1777,  @#$GDDAT ;GOOD DATA FOR ERROR TYPEOUT
4994  016372  013037  001126                    MOV     @(R0)+,@#$BDDAT ;TEST DATA
4995  016376  022737  001777  001126            CMP     #1777,  @#$BDDAT;COMPARE DATA
4996  016404  001402                            BEQ     14$             ;BRANCH IF GOOD
4997  016406  004737  016712                    JSR     PC,@#ERCLFC     ;JUMP TO ERROR FOR CLR (BIT 5)
4998                                                                    ;IN RHCS2
4999
5000                                            ;ERROR 3 REGISTER
5001
5002
5003  016412  012737  000000  001124  14$:     MOV     #0,     @#$GDDAT ;GOOD DATA FOR ERROR TYPEOUT
5004  016420  013037  001126                    MOV     @(R0)+,@#$BDDAT ;TEST DATA
5005  016424  022737  000000  001126            CMP     #0,     @#$BDDAT;COMPARE DATA
5006  016432  001402                            BEQ     15$             ;BRANCH IF GOOD
5007  016434  004737  016712                    JSR     PC,@#ERCLFC     ;JUMP TO ERROR FOR CLR (BIT 5)
5008                                                                    ;IN RHCS2
5009
```

```
                                        ;ATTENTION SUMMARY REGISTER
                                        ;
016440 013737 002004 001124 15$:  MOV   @#TOTALAT,@#$GDDAT;SET ALL BITS OF DRIVE PRESENT IN RHAS
016446 043737 002002 001124       BIC   @#ATTENT,@#$GDDAT  ;CLEAR ONLY WORKING DRIVE BIT
016454 013037 001126             MOV   @(R0)+,@#$BDDAT    ;GET RHAS
016460 123737 001124 001126       CMPB  @#$GDDAT,@#$BDDAT  ;COMPARE DATA
016466 001402                    BEQ   16$                ;BRANCH IF GOOD
016470 004737 016712             JSR   PC,@#ERCLFC        ;JUMP TO ERROR FOR CLR (BIT 5) IN RHCS2
                                        ;
                                        ;MAINTAINABILITY REGISTER
                                        ;
016474 012737 000400 001124 16$:  MOV   #400,   @#$GDDAT ;GOOD DATA FOR ERROR TYPEOUT
016502 013037 001126             MOV   @(R0)+,@#$BDDAT ;TEST DATA
016506 022737 000400 001126       CMP   #400,   @#$BDDAT;COMPARE DATA
016514 001402                    BEQ   17$              ;BRANCH IF GOOD
016516 004737 016712             JSR   PC,@#ERCLFC      ;JUMP TO ERROR FOR CLR (BIT 5)
                                                        ;IN RHCS2
                                        ;
                                        ;DRIVE STATUS REGISTER
                                        ;
016522 012737 000700 001124 17$:  MOV   #700,@#$GDDAT  ;GOOD DATA FOR PRINTOUT
016530 013046                    MOV   @(R0)+,-(SP)    ;GET RHDS1
016532 011637 001126             MOV   (SP),@#$BDDAT   ;TEST DATA
016536 042716 001000             BIC   #PROG,(SP)      ;CLEAR PROG BIT
016542 022726 000700             CMP   #700,(SP)+      ;COMPARE DATA
016546 001402                    BEQ   20$             ;BRANCH IF GOOD
016550 004737 016712             JSR   PC,@#ERCLFC     ;JUMP TO ERROR FOR DRIVE CLEAR
                                        ;
                                        ;DRIVE TYPE
                                        ;
016554 013737 001774 001124 20$:  MOV   @#SAVDT,       @#$GDDAT ;GOOD DATA FOR ERROR TYPEOUT
016562 013037 001126             MOV   @(R0)+,@#$BDDAT ;TEST DATA
016566 023737 001774 001126       CMP   @#SAVDT,       @#$BDDAT;COMPARE DATA
016574 001402                    BEQ   21$             ;BRANCH IF GOOD
016576 004737 016712             JSR   PC,@#ERCLFC     ;JUMP TO ERROR FOR CLR (BIT 5)
                                                       ;IN RHCS2
                                        ;
                                        ;SERIAL NUMBER REGISTER
                                        ;
016602 013737 001776 001124 21$:  MOV   @#SAVSN,       @#$GDDAT ;GOOD DATA FOR ERROR TYPEOUT
016610 013037 001126             MOV   @(R0)+,@#$BDDAT ;TEST DATA
016614 023737 001776 001126       CMP   @#SAVSN,       @#$BDDAT;COMPARE DATA
016622 001402                    BEQ   22$             ;BRANCH IF GOOD
016624 004737 016712             JSR   PC,@#ERCLFC     ;JUMP TO ERROR FOR CLR (BIT 5)
                                                       ;IN RHCS2
                                        ;ECC1 POSITION
                                        ;
016630 012737 000000 001124 22$:  MOV   #0,     @#$GDDAT ;GOOD DATA FOR ERROR TYPEOUT
016636 013037 001126             MOV   @(R0)+,@#$BDDAT ;TEST DATA
016642 022737 000000 001126       CMP   #0,     @#$BDDAT;COMPARE DATA
016650 001402                    BEQ   23$              ;BRANCH IF GOOD
```

```
5066  016652  004737  016712                JSR     PC,@#ERCLFC     ;JUMP TO ERROR FOR CLR (BIT 5)
5067                                                                  ;IN RHCS2
5068
5069
5070                                        ;ECC2 PATTERN
5071
5072
5073  016656  012737  000000  001124  23$:  MOV     #0,     @#$GDDAT ;GOOD DATA FOR ERROR TYPEOUT
5074  016664  013037  001126                MOV     @(R0)+,@#$BDDAT ;TEST DATA
5075  016670  022737  000000  001126        CMP     #0,     @#$BDDAT;COMPARE DATA
5076  016676  001402                        BEQ     24$             ;BRANCH IF GOOD
5077  016700  004737  016712                JSR     PC,@#ERCLFC     ;JUMP TO ERROR FOR CLR (BIT 5)
5078                                                                  ;IN RHCS2
5079
5080
5081                                        ;LOOK-AHEAD REGISTER
5082
5083  016704  005720              24$:      TST     (R0)+           ;AS THE LOOK-AHEAD REG. CANNOT BE PREDICTED
5084                                                                  ;IT IS NOT CHECKED AFTER AN INIT
5085
5086                                        ;CURRENT CYLINDER ADDRESS REGISTER
5087
5088  016706  005720              25$:      TST     (R0)+           ;AS THE CURRENT CYL REG. CANNOT BE PREDICTED
5089                                                                  ;AFTER AN INIT IT IS NOT CHECKED
5090
5091  016710                      26$:
5092
5093  016710  000410                        BR      TST42   ;BRANCH OVER JSR
5094
5095
5096  016712  014037  037534  ERCLFC: MOV     -(R0),  @#REGADR ;FAILING REGISTER ADDRESS
5097  016716  104001                ERROR   1               ;CLR FUNCTION = 10 IN RHCS1 DID
5098                                                        ;NOT CLEAR APPROPIATE BITS
5099                                                        ;OR CLEARED EXTRA BITS
5100  016720  005720                TST     (R0)+           ;UNDO -(R0) FOR ERROR
5101  016722  000207                RTS     PC              ;RETURN TO ABOVE PROGRAM
5102                                                        ;OR CLEARED EXTRA BITS
5103  016724  005720                TST     (R0)+           ;UNDO -(R0) FOR BAD DATA
5104
5105
5106  016726  004767  021166                JSR     PC,CHECKT               ;CHECK DVA, RDY, DPR, DRY
5107
5108
5109
5110
5111
5112                                ;;**************************************************************
5113                                ;*TEST 42        SEEK COMMAND TEST
5114
5115                                ;*      THE SEEK COMMAND WILL BE LOADED INTO RHCS1 WITH GO
5116                                ;*      THEN ALL REGISTERS WILL BE CHECKED
5117                                ;*      RH CLEAR WILL BE GIVEN
5118                                ;*      THEN ALL REGISTERS WILL BE CHECKED
5119
5120                                ;;**************************************************************
5121  016732  000004                TST42:  SCOPE
```

```
5122  016734  012706  001000                  MOV     #STACK,SP       ;RESET STACK
5123
5124  016740  012737  000042  004172          MOV     #TTNO,@#TSTNM   ;THIS SAVES TEST NUMBER
5125
5126
5127  016746  004737  040064                  JSR     PC,@#CLDISK     ;INIT AND SET UP GENERAL REG.
5128                                                                  ;AND UNIT NUMBER
5129  016752  012777  000001  162670          MOV     #DMD,@RHMR      ;SET DIAGNOSTIC MODE BIT
5130                                                                  ;THIS ENABLES COMMANDS WITHOUT MOL
5131                                                                  ;AND HOLDS RHLA FROM MOVING
5132
5133
5134
5135  016760  005077  162650                  CLR     @RHDST          ;MAKE DESIRED SECTOR TRACK LEGAL
5136  016764  013777  002046  162636          MOV     @#SEECOM,@RHCS1 ;LOAD SEEK COMMAND INTO RH
5137  016772  017746  162670                  MOV     @RHCC,-(SP)     ;GET RHCC
5138  016776  022726  000632                  CMP     #410.,(SP)+     ;IS CURRENT CYLINDER SAME AS 410.
5139  017002  001404                          BEQ     9$              ;BRANCH IF YES TO MAKE RHCA =409.
5140  017004  012737  000632  001210          MOV     #410.,@#$TMP5   ;GET READY TO MAKE RHCA =410
5141  017012  000403                          BR      10$             ;BRANCH TO FILL RHCA
5142  017014  012737  000631  001210  9$:     MOV     #409.,@#$TMP5   ;GET READY TO MAKE RHCA =409.
5143  017022  013777  001210  162612  10$:    MOV     @#$TMP5,@RHCA   ;MAKE DESIRED CYLINDER 401. OR 409.
5144
5145
5146                                          ;SAVE REGISTERS FOR COMPARISON AFTER GO
5147  017030  004037  040542                  JSR     R0,@#SAVER      ;SAVE
5148  017034  001622                          RHWC                    ;FROM
5149  017036  003126                          REINTO                  ;TO
5150  017040  000023                          19.                     ;NUMBER OF REGISTERS SAVED
5151
5152                                          ;GIVE GO TO COMMAND
5153  017042  052777  000001  162560          BIS     #GO,@RHCS1      ;GO TO COMMAND
5154
5155                                          ;CHANGE SAVED REGISTERS TO EXPECTED VALUES
5156
5157  017050  052737  000001  003134          BIS     #GO,@#REINTO+6  ;SAVED RHCS1
5158  017056  052737  020000  003156          BIS     #PIP,@#REINTO+30 ;SAVED RHDS1
5159  017064  042737  000200  003156          BIC     #DRY,@#REINTO+30 ;SAVED RHDS1
5160
5161
5162                                          ;AFTER GO HAS BEEN GIVEN TO SEEK COMMAND
5163                                          ;SAVE REGISTERS AGAIN SO THAT COMPARISONS CAN
5164                                          ;BE DONE
5165  017072  004037  040542                  JSR     R0,@#SAVER      ;SAVE
5166  017076  001622                          RHWC                    ;FROM
5167  017100  002062                          WRFROM                  ;TO
5168  017102  000023                          19.                     ;NUMBER OF REGISTERS SAVED
5169
5170
5171
5172                                          ;AS UPPER BYTE OF RHAS CAN BE CHANGING IN A DUAL PORT
5173                                          ;OPERATION THE UPPER BYTE OF RHAS WILL BE SAVED AS IS
5174                                          ;SO THAT THE COMPARES ARE ONLY VALID FOR THE LOWER BYTE
5175  017104  113737  003153  002107          MOVB    @#REINTO+25,@#WRFROM+25;SAVE UPPER RHAS
5176
5177
```

```
                                        ;COMPARE REGISTERS BEFORE SEEK COMMAND
                                        ;WITH AFTER GO
017112  004037  040736                  JSR     R0,@#COMPAR     ;COMPARE
017116  003126                          REINTO                  ;GOOD BUFFER
017120  002062                          WRFROM                  ;TEST BUFFER
017122  000023                          19.                     ;NUMBER
017124  017132                          1$                      ;RETURN FOR ERROR
017126  017132                          1$                      ;SAME
017130  017152                          2$                      ;RETURN FOR GOOD COMPARISON

017132  013705  044532          1$:     MOV     @#ERWORD,R5     ;GETTING READY TO INDEX
017136  060505                          ADD     R5,R5           ;DOUBLE ERROR WORD
017140  016537  001620  037534          MOV     RHWC-2(R5),@#REGADR ;FAILING REGISTER ADDRESS

017146  104001                          ERROR   1               ;IMPROPER REGISTER CHANGE
                                                                ;AFTER SEEK COMMAND
                                                                ;WITH GO IS GIVEN
017150  000207                          RTS     PC              ;RETURN TO COMPARISON


                                        ;NOW GIVE INIT AND GET GO AND PIP DOWN

017152  052712  000040          2$:     BIS     #CLR,@R2        ;RH INITILIZE
017156  013712  001760                  MOV     @#UNIT,@R2      ;REINSTATE UNIT NUMBER
017162  012777  000001  162460          MOV     #DMD,@RHMR      ;SET DIAGNOSTIC MODE BIT
                                                                ;THIS ENABLES COMMANDS WITHOUT MOL
                                                                ;AND HOLDS RHLA FROM MOVING


                                        ;CHANGE REGISTERS TO EXPECTED VALUE
017170  042737  000001  003134          BIC     #GO,@#REINTO+6  ;SAVED RHCS1
017176  042737  020000  003156          BIC     #PIP,@#REINTO+30 ;SAVED RHDS1
017204  052737  000200  003156          BIS     #DRY,@#REINTO+30 ;SAVED RHDS1
017212  017737  162446  003170          MOV     @RHLA,@#REINTO+42;SAVED RHLA
017220  013737  001210  003172          MOV     @#STMP5,@#REINTO+44 ;SAVED RHCC


                                        ;AFTER INITILIZE SAVE REGISTERS SO THAT
                                        ;COMPARES CAN BE DONE

017226  004037  040542                  JSR     R0,@#SAVER      ;SAVE
017232  001622                          RHWC                    ;FROM
017234  002062                          WRFROM                  ;TO
017236  000023                          19.                     ;NUMBER OF REGISTERS SAVED


                                        ;AS UPPER BYTE OF RHAS CAN BE CHANGING IN A DUAL PORT
                                        ;OPERATION THE UPPER BYTE OF RHAS WILL BE SAVED AS IS
                                        ;SO THAT THE COMPARES ARE ONLY VALID FOR THE LOWER BYTE
017240  113737  003153  002107          MOVB    @#REINTO+25,@#WRFROM+25;SAVE UPPER RHAS


                                        ;COMPARE REGISTERS AFTER INITILIZE
017246  004037  040736                  JSR     R0,@#COMPAR     ;COMPARE
017252  003126                          REINTO                  ;GOOD BUFFER
017254  002062                          WRFROM                  ;TEST BUFFER
017256  000023                          19.                     ;NUMBER OF REGISTERS TO BE
```

```
                                                        ;COMPARED
017260  017266                          3$              ;RETURN POINT FOR ERROR
017262  017266                          3$              ;SAME
017264  017306                          4$              ;RETURN POINT FOR GOOD COMPARISON

017266  013705  044532          3$:     MOV     @#ERWORD,R5     ;GETTING READY TO INDEX
017272  060505                  ADD     R5,R5           ;DOUBLE ERROR WORD
017274  016537  001620  037534  MOV     RHWC-2(R5),@#REGADR ;FAILING REGISTER ADDRESS
017302  104001                  ERROR   1               ;IMPROPER REGISTER
                                                        ;CONTENTS AFTER GIVING AN
                                                        ;INITILIZE FOLLOWING A
                                                        ;SEEK COMMAND
017304  000207                  RTS     PC              ;RETURN TO COMPARISON

017306                  4$:                             ;GOOD COMPARISON

017306  004737  040064          JSR     PC,@#CLDISK     ;INIT AND SET UP GENERAL REG.
                                                        ;AND UNIT NUMBER
017312  012777  000001  162330  MOV     #DMD,@RHMR      ;SET DIAGNOSTIC MODE BIT
                                                        ;THIS ENABLES COMMANDS WITHOUT MOL
                                                        ;AND HOLDS RHLA FROM MOVING

017320  013777  002046  162302  MOV     @#SEECOM,@RHCS1 ;LOAD SEEK COMMAND INTO RHC
017326  005077  162310          CLR     @RHCA           ;DESIRED CYLINDER ADDRESS
                                ;SAVE REGISTERS FOR COMPARISON AFTER GO
017332  004037  040542          JSR     R0,@#SAVER      ;SAVE
017336  001622                  RHWC                    ;FROM
017340  003126                  REINTO                  ;TO
017342  000023                  19.                     ;NUMBER OF REGISTERS SAVED

                                ;GIVE GO TO SEEK COMMAND
017344  052777  000001  162256  BIS     #GO,@RHCS1      ;GO TO SEEK COMMAND

                                ;CHANGE SAVED REGISTERS TO EXPECTED VALUES

017352  052737  000001  003134  BIS     #GO,@#REINTO+6  ;SAVED RHCS1
017360  052737  020000  003156  BIS     #PIP,@#REINTO+30 ;SAVED RHDS1
017366  042737  000200  003156  BIC     #DRY,@#REINTO+30 ;SAVED RHDS1


                                ;AFTER GO HAS BEEN GIVEN TO SEEK COMMAND
                                ;SAVE REGISTERS AGAIN SO THAT COMPARISONS CAN
                                ;BE DONE
017374  004037  040542          JSR     R0,@#SAVER      ;SAVE
017400  001622                  RHWC                    ;FROM
017402  002062                  WRFROM                  ;TO
017404  000023                  19.                     ;NUMBER OF REGISTERS SAVED


                                ;AS UPPER BYTE OF RHAS CAN BE CHANGING IN A DUAL PORT
                                ;OPERATION THE UPPER BYTE OF RHAS WILL BE SAVED AS IS
                                ;SO THAT THE COMPARES ARE ONLY VALID FOR THE LOWER BYTE
017406  113737  003153  002107  MOVB    @#REINTO+25,@#WRFROM+25;SAVE UPPER RHAS


                                ;COMPARE REGISTERS BEFORE COMMAND
```

```
                                        ;WITH AFTER GO
017414  004037  040736                  JSR     R0,@#COMPAR     ;COMPARE
017420  003126                          REINTO                  ;GOOD BUFFER
017422  002062                          WRFROM                  ;TEST BUFFER
017424  000023                          19.                     ;NUMBER
017426  017434                          5$                      ;RETURN FOR ERROR
017430  017434                          5$                      ;SAME
017432  017454                          6$                      ;RETURN FOR GOOD COMPARISON
017434  013705  044532          5$:     MOV     @#ERWORD,R5     ;GETTING READY TO INDEX
017440  060505                          ADD     R5,R5           ;DOUBLG ERROR WORD
017442  016537  001620  037534          MOV     RHWC-2(R5),@#REGADR ;FAILING REGISTER ADDRESS
017450  104001                          ERROR   1               ;IMPROPER REGISTER CHANGE
                                                                ;AFTER COMMAND
                                                                ;WITH GO IS GIVEN
017452  000207                          RTS     PC              ;RETURN TO COMPARISON

                                        ;NOW GIVE INIT AND GET GO AND PIP DOWN

017454  052712  000040          6$:     BIS     #CLR,@R2        ;RH INITILIZE
017460  013712  001760                  MOV     @#UNIT,@R2      ;REINSTATE UNIT NUMBER
017464  012777  000001  162156          MOV     #DMD,@RHMR      ;SET DIAGNOSTIC MODE BIT
                                                                ;THIS ENABLES COMMANDS WITHOUT MOL
                                                                ;AND HOLDS RHLA FROM MOVING

                                        ;CHANGE REGISTERS T EXPECTED VALUE
017472  042737  000001  003134          BIC     #GO,@#REINTO+6  ;SAVED RHCS1
017500  042737  020000  003156          BIC     #PIP,@#REINTO+30 ;SAVED RHDS1
017506  052737  000200  003156          BIS     #DRY,@#REINTO+30 ;SAVED RHDS1
017514  017737  162144  003170          MOV     @RHLA,@#REINTO+42;SAVED RHLA
017522  005037  003172                  CLR     @#REINTO+44     ;SAVED RHCC


                                        ;AFTER INITIALIZE SAVE REGISTERS SO THAT
                                        ;COMPARES CAN BE DONE
017526  004037  040542                  JSR     R0,@#SAVER      ;SAVE
017532  001622                          RHWC                    ;FROM
017534  002062                          WRFROM                  ;TO
017536  000023                          19.                     ;NUMBER OF REGISTERS SAVED

                                        ;AS UPPER BYTE OF RHAS CAN BE CHANGING IN A DUAL PORT
                                        ;OPERATION THE UPPER BYTE OF RHAS WILL BE SAVED AS IS
                                        ;SO THAT THE COMPARES ARE ONLY VALID FOR THE LOWER BYTE
017540  113737  003153  002107          MOVB    @#REINTO+25,@#WRFROM+25;SAVE UPPER RHAS


                                        ;COMPARE REGISTERS AFTER INITALIZE
017546  004037  040736                  JSR     R0,@#COMPAR     ;COMPARE
017552  003126                          REINTO                  ;GOOD BUFFER
017554  002062                          WRFROM                  ;TEST BUFFER
017556  000023                          19.                     ;NUMBER OF REGISTERS TO BE
                                                                ;COMPARED
017560  017566                          7$                      ;RETURN POINT FOR ERROR
017562  017566                          7$                      ;SAME
017564  017606                          8$                      ;RETURN POINT FOR GOOD COMPARISON

017566  013705  044532          7$:     MOV     @#ERWORD,R5     ;GETTING READY TO INDEX
```

```
5346  017572  060505                          ADD     R5,R5           ;DOUBLE ERROR WORD
5347  017574  016537  001620  037534          MOV     RHWC-2(R5),@#REGADR ;FAILING REGISTER ADDRESS
5348  017602  104001                          ERROR   1               ;# CONTENTS AFTER GIVING AN
5349                                                                  ;  COMMAND
5350  017604  000207                          RTS     PC              ;RETURN TO COMPARISON
5351  017606                          8$:                             ;GOOD COMPARISON
5352
5353
5354                                  ;;**********************************************************************
5355                                  ;*TEST 43        UNLOAD COMMAND TEST
5356
5357                                  ;*      THE UNLOAD COMMAND WILL BE LOADED INTO RHCS1 WITH GO
5358                                  ;*      THEN ALL REGISTERS WILL BE CHECKED
5359                                  ;*      RH CLEAR WILL BE GIVEN
5360
5361                                  ;;**********************************************************************
5362  017606  000004                  TST43:  SCOPE
5363  017610  012706  001000                  MOV     #STACK,SP       ;RESET STACK
5364
5365  017614  012737  000043  004172          MOV     #TTNO,@#TSTNM   ;THIS SAVES TEST NUMBER
5366
5367  017622  004737  040064                  JSR     PC,@#CLDISK     ;INIT AND SET UP GENERAL REG.
5368                                                                  ;AND UNIT NUMBER
5369  017626  012777  000001  162014          MOV     #DMD,@RHMR      ;SET DIAGNOSTIC MODE BIT
5370                                                                  ;THIS ENABLES COMMANDS WITHOUT MOL
5371                                                                  ;AND HOLDS RHLA FROM MOVING
5372
5373  017634  013777  002020  161766          MOV     @#UNLOAD,@RHCS1 ;LOAD UNLOAD COMMAND INTO RH
5374
5375                                  ;SAVE REGISTERS FOR COMPARISON AFTER GO
5376  017642  004037  040542                  JSR     R0,@#SAVER      ;SAVE
5377  017646  001622                          RHWC                    ;FROM
5378  017650  003126                          REINTO                  ;TO
5379  017652  000023                          19.                     ;NUMBER OF REGISTERS SAVED
5380
5381                                  ;GIVE GO TO UNLOAD COMMAND
5382  017654  052777  000001  161746          BIS     #GO,@RHCS1      ;GO TO UNLOAD COMMAND
5383                                  ;CHANGE SAVED REGISTERS TO EXPECTED VALUES
5384  017662  052737  000001  003134          BIS     #GO,@#REINTO+6  ;SAVED RHCS1
5385  017670  052737  020000  003156          BIS     #PIP,@#REINTO+30 ;SAVED RHDS1
5386  017676  042737  000200  003156          BIC     #DRY,@#REINTO+30 ;SAVED RHDS1
5387
5388  017704  005737  001100                  TST     @#$PASS ;IS THIS FIRST PASS
5389  017710  001053                          BNE     5$              ;BRANCH IF NOT FIRST PASS
5390  017712  032737  020000  001140          BIT     #SW13,@#SWR     ;INHIBIT ERROR PRINT HIGH?
5391  017720  001047                          BNE     5$              ;BRANCH IF SW13 HIGH
5392  017722  104400  017730                  TYPE    ,65$            ;;TYPE ASCIZ STRING
5393  017726  000441                          BR      64$             ;;GET OVER THE ASCIZ
5394                                  ;;65$:  .ASCIZ  <15><12>/IF DRIVE CONNECTED "STAND BY" LAMP SHOULD BE LIT ON DRIVE NO  /
5395  020032                          64$:
5396  020032  013746  001760                  MOV     @#UNIT,-(SP)    ;UNIT UNDER TEST
5397  020036  104404                          TYPDS
5398
5399                                  ;AFTER GO HAS BEEN GIVEN TO UNLOAD COMMAND
5400                                  ;SAVED REGISTERS AGAIN SO THAT COMPARISONS CAN
5401                                  ;BE DONE
```

```
 5402  020040  004037  040542          5$:   JSR     R0,@#SAVER      ;SAVE
 5403  020044  001622                        RHWC                    ;FROM
 5404  020046  002062                        WRFROM                  ;TO
 5405  020050  000023                        19.                     ;NUMBER OF REGISTERS SAVED
 5406
 5407
 5408                                        ;AS UPPER BYTE OF RHAS CAN BE CHANGING IN A DUAL PORT
 5409                                        ;OPERATION THE UPPER BYTE OF RHAS WILL BE SAVED AS IS
 5410                                        ;SO THAT THE COMPARES ARE ONLY VALID FOR THE LOWER BYTE
 5411  020052  113737  003153  002107        MOVB    @#REINTO+25,@#WRFROM+25;SAVE UPPER RHAS
 5412
 5413
 5414                                        ;COMPARE REGISTERS BEFORE UNLOAD COMMAND
 5415                                        ;WITH AFTER GO
 5416  020060  004037  040736                JSR     R0,@#COMPAR     ;COMPARE
 5417  020064  003126                        REINTO                  ;GOOD BUFFER
 5418  020066  002062                        WRFROM                  ;TEST BUFFER
 5419  020070  000023                        19.                     ;NUMBER
 5420  020072  020100                        1$                      ;RETURN FOR ERROR
 5421  020074  020100                        1$                      ;SAME
 5422  020076  020120                        2$                      ;RETURN FOR GOOD COMPARISON
 5423  020100  013705  044532          1$:   MOV     @#ERWORD,R5     ;GETTING READY TO INDEX
 5424  020104  060505                        ADD     R5,R5           ;DOUBLE ERROR WORD
 5425  020106  016537  001620  037534        MOV     RHWC-2(R5),@#REGADR ;FAILING REGISTER ADDRESS
 5426
 5427  020114  104001                        ERROR   1               ;IMPROPER REGISTER CHANGE
 5428                                                                ;AFTER UNLOAD COMMAND
 5429                                                                ;WITH GO IS GIVEN
 5430  020116  000207                        RTS     PC              ;RETURN TO COMPARISON
 5431
 5432                                        ;NOW GIVE INIT AND GET ALL GO AND PIP DOWN
 5433  020120  052712  000040          2$:   BIS     #CLR,@R2        ;RH INITILIZE
 5434  020124  013712  001760                MOV     @#UNIT,@R2      ;REINSTATE UNIT NUMBER
 5435  020130  012777  000001  161512        MOV     #DMD,@RHMR      ;SET DIAGNOSTIC MODE BIT
 5436                                                                ;THIS ENABLES COMMANDS WITHOUT MOL
 5437                                                                ;AND HOLDS RHLA FROM MOVING
 5438
 5439                                        ;CHANGE REGISTERS TO EXPECTED VALUE
 5440  020136  042737  000001  003134        BIC     #GO,@#REINTO+6  ;SAVED RHCS1
 5441  020144  042737  020000  003156        BIC     #PIP,@#REINTO+30 ;SAVED RHDS1
 5442  020152  052737  000200  003156        BIS     #DRY,@#REINTO+30 ;SAVED RHDS1
 5443  020160  017737  161500  003170        MOV     @RHLA,@#REINTO+42;SAVED RHLA
 5444
 5445                                        ;AFTER INITIALIZE SAVE REGISTERS SO THAT
 5446                                        ;COMPARES CAN BE DONE
 5447  020166  004037  040542                JSR     R0,@#SAVER      ;SAVE
 5448  020172  001622                        RHWC                    ;FROM
 5449  020174  002062                        WRFROM                  ;TO
 5450  020176  000023                        19.                     ;NUMBER OF REGISTERS SAVED
 5451
 5452                                        ;COMPARE REGISTERS AFTER INITIALIZE
 5453                                        ;AS UPPER BYTE OF RHAS CAN BE CHANGING IN A DUAL PORT
 5454                                        ;OPERATION THE UPPER BYTE OF RHAS WILL BE SAVED AS IS
 5455                                        ;SO THAT THE COMPARES ARE ONLY VALID FOR THE LOWER BYTE
 5456  020200  113737  003153  002107        MOVB    @#REINTO+25,@#WRFROM+25;SAVE UPPER RHAS
 5457
```

```
5458
5459  020206  004037  040736                  JSR     R0,@#COMPAR     ;COMPARE
5460  020212  003126                          REINTO                  ;GOOD BUFFER
5461  020214  002062                          WRFROM                  ;TEST BUFFER
5462  020216  000023                          19.                     ;NUMBER OF REGISTERS TO BE
5463                                                                  ;COMPARED
5464  020220  020226                          3$                      ;RETURN POINT FOR ERROR
5465  020222  020226                          3$                      ;SAME
5466  020224  020246                          4$                      ;RETURN POINT FOR GOOD COMPARISON
5467
5468  020226  013705  044532          3$:     MOV     @#ERWORD,R5     ;GETTING READY TO INDEX
5469  020232  060505                          ADD     R5,R5           ;DOUBLE ERROR WORD
5470  020234  016537  001620  037534          MOV     RHWC-2(R5),@#REGADR ;FAILING REGISTER ADDRESS
5471  020242  104001                          ERROR   1               ;IMPROPER REGISTER
5472                                                                  ;CONTENTS AFTER GIVING AN
5473                                                                  ;UNLOAD COMMAND
5474  020244  000207                          RTS     PC              ;RETURN TO COMPARISON
5475  020246                          4$:                             ;GOOD COMPARISON
```

```
                                ;;*****************************************************************
                                ;*TEST 44       OFFSET COMMAND TEST

                                ;*      THE OFFSET COMMAND WILL BE LOADED INTO RHCS1 WITH GO
                                ;*      THEN ALL REGISTERS WILL BE CHECKED
                                ;*      RH CLEAR WILL BE GIVEN
                                ;*      THEN ALL REGISTERS WILL BE CHECKED

                                ;;*****************************************************************
020246  000004                  TST44:  SCOPE
020250  012706  001000                  MOV     #STACK,SP       ;RESET STACK

020254  012737  000044  004172          MOV     #TTNO,@#TSTNM   ;THIS SAVES TEST NUMBER

020262  004737  040064                  JSR     PC,@#CLDISK     ;INIT AND SET UP GENERAL REG.
                                                                ;AND UNIT NUMBER
020266  012777  000001  161354          MOV     #DMD,@RHMR      ;SET DIAGNOSTIC MODE BIT
                                                                ;THIS ENABLES COMMANDS WITHOUT MOL
                                                                ;AND HOLDS RHLA FROM MOVING
                                        ;GIVE ONE INDEX PULSE TO CLEAR RHLA BEFORE THE START OF THIS TEST
020274  052777  000004  161346          BIS     #MINX,@RHMR     ;SET INDEX PULSE
020302  042777  000004  161340          BIC     #MINX,@RHMR     ;CLEAR INDEX


                                ;TO ENABLE LOOP ON THIS TEST THE POSITIONER HAS TO
                                ;BE BROUGHT TO CENTER LINE

020310  017777  161352  161324          MOV     @RHCC,@RHCA     ;SET DESIRED CYLINDER TO RHCC
020316  013711  002046                  MOV     @#SEECOM,@R1    ;SEEK COMMAND TO RHCS1
020322  005211                          INC     @R1             ;GO TO SEEK COMMAND

                                ;FOUR SECTOR CLOCKS ARE GIVEN TO TAKE POSITIONER OFF OFFSET POSITION
020324  012700  000004                  MOV     #4,R0           ;COUNTER
020330  012777  000011  161312  5$:     MOV     #MSTCK!DMD,@RHMR        ;SET SECTOR CLOCK
020336  012777  000001  161304          MOV     #DMD,@RHMR      ;RESET SECTOR CLOCK
020344  005300                          DEC     R0              ;COUNT
020346  001370                          BNE     5$              ;BRANCH IF NOT COMPLETE

020350  004737  040064                  JSR     PC,@#CLDISK     ;INIT AND SET UP GENERAL REG.
                                                                ;AND UNIT NUMBER
020354  012777  000001  161266          MOV     #DMD,@RHMR      ;SET DIAGNOSTIC MODE BIT
                                                                ;THIS ENABLES COMMANDS WITHOUT MOL
                                                                ;AND HOLDS RHLA FROM MOVING

020362  013777  002050  161240          MOV     @#OFSETC,@RHCS1 ;LOAD AN OFFSET BIT
020370  012777  000001  161242          MOV     #OF25,@RHOF     ;SET AN OFFSET BIT
                                        ;SAVE REGISTERS FOR COMPARISON AFTER GO
020376  004037  040542                  JSR     R0,@#SAVER      ;SAVE
020402  001622                          RHWC                    ;FROM
```

```
5532  020404  003126                          REINTO                  ;TO
5533  020406  000023                          19.                     ;NUMBER OF REGISTERS SAVED
5534
5535                                          ;GIVE GO TO OFFSET COMMAND
5536  020410  052777  000001  161212          BIS     #GO,@RHCS1      ;GO TO OFFSET COMMAND
5537
5538                                          ;CHANGE SAVED REGISTERS TO EXPECTED VALUES
5539  020416  052737  000001  003134          BIS     #GO,@#REINTO+6  ;SAVED RHCS1
5540  020424  052737  020000  003156          BIS     #PIP,@#REINTO+30 ;SAVED RHDS1
5541  020432  042737  000200  003156          BIC     #DRY,@#REINTO+30 ;SAVED RHDS1
5542
5543                                          ;AFTER GO HAS BEEN GIVEN TO OFFSET COMMAND
5544                                          ;SAVE REGISTERS AGAIN SO THAT COMPARISONS CAN
5545                                          ;BE DONE
5546  020440  004037  040542                  JSR     R0,@#SAVER      ;SAVE
5547  020444  001622                          RHWC                    ;FROM
5548  020446  002062                          WRFROM                  ;TO
5549  020450  000023                          19.                     ;NUMBER OF REGISTERS SAVED
5550
5551
5552                                          ;AS UPPER BYTE OF RHAS CAN BE CHANGING IN A DUAL PORT
5553                                          ;OPERATION THE UPPER BYTE OF RHAS WILL BE SAVED AS IS
5554                                          ;SO THAT THE COMPARES ARE ONLY VALID FOR THE LOWER BYTE
5555  020452  113737  003153  002107          MOVB    @#REINTO+25,@#WRFROM+25;SAVE UPPER RHAS
5556
5557
5558                                          ;COMPARE REGISTERS BEFORE OFFSET COMMAND
5559                                          ;WITH AFTER GO
5560  020460  004037  040736                  JSR     R0,@#COMPAR     ;COMPARE
5561  020464  003126                          REINTO                  ;GOOD BUFFER
5562  020466  002062                          WRFROM                  ;TEST BUFFER
5563  020470  000023                          19.                     ;NUMBER
5564  020472  020500                          1$                      ;RETURN FOR ERROR
5565  020474  020500                          1$                      ;SAME
5566  020476  020520                          2$                      ;RETURN FOR GOOD COMPARISON
5567
5568  020500  013705  044532          1$:     MOV     @#ERWORD,R5     ;GETTING READY TO INDEX
5569  020504  060505                          ADD     R5,R5           ;DOUBLE ERROR WORD
5570  020506  016537  001620  037534          MOV     RHWC-2(R5),@#REGADR ;FAILING REGISTER ADDRESS
5571  020514  104001                          ERROR   1               ;IMPROPER REGISTER CHANGE
5572                                                                  ;AFTER OFFSET COMMAND
5573                                                                  ;WITH GO IS GIVEN
5574  020516  000207                          RTS     PC              ;RETURN TO COMPARISON
5575
5576                                          ;NOW GIVE INIT AND GET ALL GO AND PIP DOWN
5577
5578  020520  052712  000040          2$:     BIS     #CLR,@R2        ;RH INITILIZE
5579  020524  013712  001760                  MOV     @#UNIT,@R2      ;REINSTATE UNIT NUMBER
5580  020530  012777  000001  161112          MOV     #DMD,@RHMR      ;SET DIAGNOSTIC MODE BIT
5581                                                                  ;THIS ENABLES COMMANDS WITHOUT MOL
5582                                                                  ;AND HOLDS RHLA FROM MOVING
5583
5584                                          ;CHANGE REGISTERS TO EXPECTED VALUE
5585  020536  042737  000001  003134          BIC     #GO,@#REINTO+6  ;SAVED RHCS1
5586  020544  042737  000001  003144          BIC     #0F25,@#REINTO+16;SAVED RHOF
5587  020552  042737  020000  003156          BIC     #PIP,@#REINTO+30 ;SAVED RHDS1
```

```
5588  020560  052737  000200  003156          BIS     #DRY,@#REINTO+30 ;SAVED RHDS1
5589  020566  017737  161072  003170          MOV     @RHLA,@#REINTO+42;SAVED RHLA
5590
5591                                          ;AFTER INITIALIZE SAVE REGISTERS SO THAT
5592                                          ;COMPARES CAN BE DONE
5593  020574  004037  040542                  JSR     R0,@#SAVER      ;SAVE
5594  020600  001622                          RHWC                    ;FROM
5595  020602  002062                          WRFROM                  ;TO
5596  020604  000023                          19.                     ;NUMBER OF REGISTERS SAVED
5597
5598                                          ;AS UPPER BYTE OF RHAS CAN BE CHANGING IN A DUAL PORT
5599                                          ;OPERATION THE UPPER BYTE OF RHAS WILL BE SAVED AS IS
5600                                          ;SO THAT THE COMPARES ARE ONLY VALID FOR THE LOWER BYTE
5601  020606  113737  003153  002107          MOVB    @#REINTO+25,@#WRFROM+25;SAVE UPPER RHAS
5602
5603
5604                                          ;COMPARE REGISTERS AFTER INITIALIZE
5605  020614  004037  040736                  JSR     R0,@#COMPAR     ;COMPARE
5606  020620  003126                          REINTO                  ;GOOD BUFFER
5607  020622  002062                          WRFROM                  ;TEST BUFFER
5608  020624  000023                          19.                     ;NUMBER OF REGISTERS TO BE
5609                                                                  ;COMPARED
5610  020626  020634                          3$                      ;RETURN POINT FOR ERROR
5611  020630  020634                          3$                      ;SAME
5612  020632  020654                          4$                      ;RETURN POINT FOR GOOD COMPARISON
5613
5614  020634  013705  044532          3$:     MOV     @#ERWORD,R5     ;GETTING READY TO INDEX
5615  020640  060505                          ADD     R5,R5           ;DOUBLE ERROR WORD
5616  020642  016537  001620  037534          MOV     RHWC-2(R5),@#REGADR ;FAILING REGISTER ADDRESS
5617  020650  104001                          ERROR   1               ;IMPROPER REGISTER
5618                                                                  ;CONTENTS AFTER GIVING AN
5619                                                                  ;INITIALIZE FOLLOWING A
5620                                                                  ;OFFSET COMMAND
5621  020652  000207                          RTS     PC              ;RETURN TO COMPARISON
5622
5623  020654                          4$:                             ;GOOD COMPARISON
5624
5625
5626                                          ;;*****************************************************************
5627                                          ;*TEST 45       RETURN TO CENTER LINE COMMAND TEST
5628
5629                                          ;*      THE RETURN TO CENTER LINE COMMAND WILL BE LOADED INTO RHCS1 WITH GO
5630                                          ;*      THEN ALL REGISTERS WILL BE CHECKED
5631                                          ;*      RH CLEAR WILL BE GIVEN
5632                                          ;*      THEN ALL REGISTERS WILL BE CHECKED
5633
5634                                          ;;*****************************************************************
5635  020654  000004                  TST45:  SCOPE
5636  020656  012706  001000                  MOV     #STACK,SP       ;RESET STACK
5637
5638  020662  012737  000045  004172          MOV     #TTNO,@#TSTNM   ;THIS SAVES TEST NUMBER
5639
5640
5641  020670  004737  040064                  JSR     PC,@#CLDISK     ;INIT AND SET UP GENERAL REG.
5642                                                                  ;AND UNIT NUMBER
5643  020674  012777  000001  160746          MOV     #DMD,@RHMR      ;SET DIAGNOSTIC MODE BIT
```

```
                                                        ;THIS ENABLES COMMANDS WITHOUT MOL
                                                        ;AND HOLDS RHLA FROM MOVING

                                                ;GIVE ONE INDEX PULSE TO CLEAR RHLA BEFORE THE START OF THIS TEST
020702  052777  000004  160740                BIS     #MINX,@RHMR     ;SET INDEX PULSE
020710  042777  000004  160732                BIC     #MINX,@RHMR     ;CLEAR INDEX


020716  013777  002052  160704                MOV     @#RETCL,@RHCS1  ;LOAD RETURN TO CENTER LINE COMMAND INTO RHCS1

                                                ;SAVE REGISTERS FOR COMPARISON AFTER GO
020724  004037  040542                        JSR     R0,@#SAVER      ;SAVE
020730  001622                                RHWC                    ;FROM
020732  003126                                REINTO                  ;TO
020734  000023                                19.                     ;NUMBER OF REGISTERS SAVED

                                                ;GIVE GO TO RETURN TO CENTER LINE COMMAND
020736  052777  000001  160664                BIS     #GO,@RHCS1      ;GO TO RETURN TO CENTER COMMAND


                                                ;FOUR SECTOR CLOCKS ARE GIVEN TO TAKE POSITIONER TO CENTER LINE
020744  012700  000004                        MOV     #4,R0           ;COUNTER
020750  012777  000011  160672  5$:           MOV     #MSTCK!DMD,@RHMR        ;SET SECTOR CLOCK
020756  012777  000001  160664                MOV     #DMD,@RHMR      ;RESET SECTOR CLOCK
020764  005300                                DEC     R0              ;COUNT
020766  001370                                BNE     5$              ;BRANCH IF NOT COMPLETE


                                                ;CHANGE SAVED REGISTERS TO EXPECTED VALUES
020770  052737  000001  003134                BIS     #GO,@#REINTO+6  ;SAVED RHCS1
020776  052737  020000  003156                BIS     #PIP,@#REINTO+30 ;SAVED RHDS1
021004  042737  000200  003156                BIC     #DRY,@#REINTO+30 ;SAVED RHDS1


                                                ;AFTER GO HAS BEEN GIVEN TO RETURN TO CENTER LINE COMMAND
                                                ;SAVE REGISTERS AGAIN SO THAT COMPARISONS CAN
                                                ;BE DONE
021012  004037  040542                        JSR     R0,@#SAVER      ;SAVE
021016  001622                                RHWC                    ;FROM
021020  002062                                WRFROM                  ;TO
021022  000023                                19.                     ;NUMBER OF REGISTERS SAVED


                                                ;AS UPPER BYTE OF RHAS CAN BE CHANGING IN A DUAL PORT
                                                ;OPERATION THE UPPER BYTE OF RHAS WILL BE SAVED AS IS
                                                ;SO THAT THE COMPARES ARE ONLY VALID FOR THE LOWER BYTE
021024  113737  003153  002107                MOVB    @#REINTO+25,@#WRFROM+25;SAVE UPPER RHAS


                                                ;COMPARE REGISTERS BEFORE RETURN TO CENTER LINE COMMAND
                                                ;WITH AFTER GO
021032  004037  040736                        JSR     R0,@#COMPAR     ;COMPARE
021036  003126                                REINTO                  ;GOOD BUFFER
021040  002062                                WRFROM                  ;TEST BUFFER
021042  000023                                19.                     ;NUMBER
021044  021052                                1$                      ;RETURN FOR ERROR
```

```
                                        1$                      ;SAME
                                        2$                      ;RETURN FOR GOOD COMPARISON

                        1$:     MOV     @#ERWORD,R5             ;GETTING READY TO INDEX
                        ADD     R5,R5                           ;DOUBLE ERROR WORD
                        MOV     RHWC-2(R5),@#REGADR ;FAILING REGISTER ADDRESS
                        ERROR   1                               ;IMPROPER REGISTER CHANGE
                                                                ;AFTER RETURN TO CENTER LINE COMMAND
                                                                ;WITH GO IS GIVEN
                        RTS     PC                              ;RETURN TO COMPARISON

                        ;NOW GIVE INIT AND GET ALL GO AND PIP DOWN

                        2$:     BIS     #CLR,@R2                ;RH INITILIZE
                        MOV     @#UNIT,@R2                      ;REINSTATE UNIT NUMBER
                        MOV     #DMD,@RHMR                      ;SET DIAGNOSTIC MODE BIT
                                                                ;THIS ENABLES COMMANDS WITHOUT MOL
                                                                ;AND HOLDS RHLA FROM MOVING

                        ;CHANGE REGISTERS TO EXPECTED VALUE
                        BIC     #GO,@#REINTO+6  ;SAVED RHCS1
                        BIC     #PIP,@#REINTO+30 ;SAVED RHDS1
                        BIS     #DRY,@#REINTO+30 ;SAVED RHDS1
                        MOV     @RHLA,@#REINTO+42;SAVED RHLA

                        ;AFTER INITIALIZE SAVE REGISTERS SO THAT
                        ;COMPARES CAN BE DONE
                        JSR     R0,@#SAVER              ;SAVE
                        RHWC                            ;FROM
                        WRFROM                          ;TO
                        19.                             ;NUMBER OF REGISTERS SAVED

                        ;AS UPPER BYTE OF RHAS CAN BE CHANGING IN A DUAL PORT
                        ;OPERATION THE UPPER BYTE OF RHAS WILL BE SAVED AS IS
                        ;SO THAT THE COMPARES ARE ONLY VALID FOR THE LOWER BYTE
                        MOVB    @#REINTO+25,@#WRFROM+25;SAVE UPPER RHAS

                        ;COMPARE REGISTERS AFTER INITIALIZE
                        JSR     R0,@#COMPAR             ;COMPARE
                        REINTO                          ;GOOD BUFFER
                        WRFROM                          ;TEST BUFFER
                        19.                             ;NUMBER OF REGISTERS TO BE
                                                        ;COMPARED
                        3$                              ;RETURN POINT FOR ERROR
                        3$                              ;SAME
                        4$                              ;RETURN POINT FOR GOOD COMPARISON

                        3$:     MOV     @#ERWORD,R5             ;GETTING READY TO INDEX
                        ADD     R5,R5                           ;DOUBLE ERROR WORD
                        MOV     RHWC-2(R5),@#REGADR ;FAILING REGISTER ADDRESS
                        ERROR   1                               ;IMPROPER REGISTER
                                                                ;CONTENTS AFTER GIVING AN
                                                                ;INITIALIZE FOLLOWING A RETURN TO
                                                                ;CENTER LINE COMMAND
                        RTS     PC                              ;RETURN TO COMPARISON
```

```
5756
5757  021220                    4$:                         ;GOOD COMPARISON
```

```
                                ;;*****************************************************************
                                ;*TEST 46       RECALIBRATE COMMAND TEST

                                ;*      THE RECALIBRATE COMMAND WILL BE LOADED INTO RHCS1 WITH GO
                                ;*      THEN ALL REGISTERS WILL BE CHECKED
                                ;*      RH CLEAR WILL BE GIVEN
                                ;*      THEN ALL REGISTERS WILL BE CHECKED

                                ;;*****************************************************************
021220  000004                  TST46:  SCOPE
021222  012706  001000                  MOV     #STACK,SP       ;RESET STACK

021226  012737  000046  004172          MOV     #TTNO,@#TSTNM   ;THIS SAVES TEST NUMBER

021234  004737  040064                  JSR     PC,@#CLDISK     ;INIT AND SET UP GENERAL REG.
                                                                ;AND UNIT NUMBER
021240  012777  000001  160402          MOV     #DMD,@RHMR      ;SET DIAGNOSTIC MODE BIT
                                                                ;THIS ENABLES COMMANDS WITHOUT MOL
                                                                ;AND HOLDS RHLA FROM MOVING


                                        ;GIVE ONE INDEX PULSE TO CLEAR RHLA BEFORE THE START OF THIS TEST
021246  052777  000004  160374          BIS     #MINX,@RHMR     ;SET INDEX PULSE
021254  042777  000004  160366          BIC     #MINX,@RHMR     ;CLEAR INDEX


021262  013777  002022  160340          MOV     @#RECALI,@RHCS1 ;LOAD RECALIBRATE COMMAND INTO RHCS1

                                        ;SAVE REGISTERS FOR COMPARISON AFTER GO
021270  004037  040542                  JSR     R0,@#SAVER      ;SAVE
021274  001622                          RHWC                    ;FROM
021276  003126                          REINTO                  ;TO
021300  000023                          19.                     ;NUMBER OF REGISTERS SAVED

                                        ;GIVE GO TO RECALIBRATE COMMAND
021302  052777  000001  160320          BIS     #GO,@RHCS1      ;GO TO RECALIBRATE COMMAND


                                        ;FOUR SECTOR CLOCKS ARE GIVEN TO TAKE POSITIONER TO CYLINDER 0
021310  012700  000004                  MOV     #4,R0           ;COUNTER
021314  012777  000011  160326  5$:     MOV     #MSTCK!DMD,@RHMR        ;SET SECTOR CLOCK
021322  012777  000001  160320          MOV     #DMD,@RHMR      ;RESET SECTOR CLOCK
021330  005300                          DEC     R0              ;COUNT
021332  001370                          BNE     5$              ;BRANCH IF NOT COMPLETE


                                        ;CHANGE SAVED REGISTERS TO EXPECTED VALUES
021334  052737  000001  003134          BIS     #GO,@#REINTO+6  ;SAVED RHCS1
021342  052737  020000  003156          BIS     #PIP,@#REINTO+30 ;SAVED RHDS1
```

```
5814  021350  042737  000200  003156          BIC     #DRY,@#REINTO+30 ;SAVED RHDS1
5815
5816                                          ;AFTER GO HAS BEEN GIVEN TO RECALIBRATE COMMAND
5817                                          ;SAVE REGISTERS AGAIN SO THAT COMPARISONS CAN
5818                                          ;BE DONE
5819  021356  004037  040542                  JSR     R0,@#SAVER      ;SAVE
5820  021362  001622                          RHWC                    ;FROM
5821  021364  002062                          WRFROM                  ;TO
5822  021366  000023                          19.                     ;NUMBER OF REGISTERS SAVED
5823
5824                                          ;AS UPPER BYTE OF RHAS CAN BE CHANGING IN A DUAL PORT
5825                                          ;OPERATION THE UPPER BYTE OF RHAS WILL BE SAVED AS IS
5826                                          ;SO THAT THE COMPARES ARE ONLY VALID FOR THE LOWER BYTE
5827  021370  113737  003153  002107          MOVB    @#REINTO+25,@#WRFROM+25;SAVE UPPER RHAS
5828
5829
5830                                          ;COMPARE REGISTERS BEFORE RECALIBRATE COMMAND
5831                                          ;WITH AFTER GO
5832  021376  004037  040736                  JSR     R0,@#COMPAR     ;COMPARE
5833  021402  003126                          REINTO                  ;GOOD BUFFER
5834  021404  002062                          WRFROM                  ;TEST BUFFER
5835  021406  000023                          19.                     ;NUMBER
5836  021410  021416                          1$                      ;RETURN FOR ERROR
5837  021412  021416                          1$                      ;SAME
5838  021414  021436                          2$                      ;RETURN FOR GOOD COMPARISON
5839
5840  021416  013705  044532          1$:     MOV     @#ERWORD,R5     ;GETTING READY TO INDEX
5841  021422  060505                          ADD     R5,R5           ;DOUBLE ERROR WORD
5842  021424  016537  001620  037534          MOV     RHWC-2(R5),@#REGADR ;FAILING REGISTER ADDRESS
5843  021432  104001                          ERROR   1               ;IMPROPER REGISTER CHANGE
5844                                                                  ;AFTER RECALIBRATE COMMAND
5845                                                                  ;WITH GO IS GIVEN
5846  021434  000207                          RTS     PC              ;RETURN TO COMPARISON
5847
5848                                          ;NOW GIVE INIT AND GET ALL GO AND PIP DOWN
5849
5850  021436  052712  000040          2$:     BIS     #CLR,@R2        ;RH INITILIZE
5851  021442  013712  001760                  MOV     @#UNIT,@R2      ;REINSTATE UNIT NUMBER
5852  021446  012777  000001  160174          MOV     #DMD,@RHMR      ;SET DIAGNOSTIC MODE BIT
5853                                                                  ;THIS ENABLES COMMANDS WITHOUT MOL
5854                                                                  ;AND HOLDS RHLA FROM MOVING
5855
5856                                          ;CHANGE REGISTERS TO EXPECTED VALUE
5857  021454  042737  000001  003134          BIC     #GO,@#REINTO+6  ;SAVED RHCS1
5858  021462  042737  020000  003156          BIC     #PIP,@#REINTO+30 ;SAVED RHDS1
5859  021470  052737  000200  003156          BIS     #DRY,@#REINTO+30 ;SAVED RHDS1
5860  021476  017737  160162  003170          MOV     @RHLA,@#REINTO+42;SAVED RHLA
5861
5862                                          ;AFTER INITIALIZE SAVE REGISTERS SO THAT
5863                                          ;COMPARES CAN BE DONE
5864  021504  004037  040542                  JSR     R0,@#SAVER      ;SAVE
5865  021510  001622                          RHWC                    ;FROM
5866  021512  002062                          WRFROM                  ;TO
5867  021514  000023                          19.                     ;NUMBER OF REGISTERS SAVED
5868
5869                                          ;AS UPPER BYTE OF RHAS CAN BE CHANGING IN A DUAL PORT
```

```
                                        ;OPERATION THE UPPER BYTE OF RHAS WILL BE SAVED AS IS
                                        ;SO THAT THE COMPARES ARE ONLY VALID FOR THE LOWER BYTE
021516  113737  003153  002107        MOVB    @#REINTO+25,@#WRFROM+25;SAVE UPPER RHAS


                                        ;COMPARE REGISTERS AFTER INITIALIZE
021524  004037  040736                JSR     R0,@#COMPAR     ;COMPARE
021530  003126                        REINTO                  ;GOOD BUFFER
021532  002062                        WRFROM                  ;TEST BUFFER
021534  000023                        19.                     ;NUMBER OF REGISTERS TO BE
                                                                ;COMPARED
021536  021544                        3$                      ;RETURN POINT FOR ERROR
021540  021544                        3$                      ;SAME
021542  021564                        4$                      ;RETURN POINT FOR GOOD COMPARISON

021544  013705  044532        3$:     MOV     @#ERWORD,R5     ;GETTING READY TO INDEX
021550  060505                        ADD     R5,R5           ;DOUBLE ERROR WORD
021552  016537  001620  037534        MOV     RHWC-2(R5),@#REGADR ;FAILING REGISTER ADDRESS
021560  104001                        ERROR   1               ;IMPROPER REGISTER
                                                                ;CONTENTS AFTER GIVING AN
                                                                ;INITIALIZE FOLLOWING A
                                                                ;RECALIBRATE COMMAND
021562  000207                        RTS     PC              ;RETURN TO COMPARISON

021564                          4$:                             ;GOOD COMPARISON


                                        ;;**************************************************************
                                        ;*TEST 47       RELEASE COMMAND TEST

                                        ;*      THE RELEASE COMMAND WILL BE LOADED INTO RHCS1 WITH GO
                                        ;*      THEN ALL REGISTERS WILL BE CHECKED
                                        ;*      RH CLEAR WILL BE GIVEN
                                        ;*      THEN ALL REGISTERS WILL BE CHECKED

                                        ;;**************************************************************
021564  000004                TST47:  SCOPE
021566  012706  001000                MOV     #STACK,SP       ;RESET STACK

021572  012737  000047  004172        MOV     #TTNO,@#TSTNM   ;THIS SAVES TEST NUMBER


021600  004737  040064                JSR     PC,@#CLDISK     ;INIT AND SET UP GENERAL REG.
                                                                ;AND UNIT NUMBER
021604  012777  000001  160036        MOV     #DMD,@RHMR      ;SET DIAGNOSTIC MODE BIT
                                                                ;THIS ENABLES COMMANDS WITHOUT MOL
                                                                ;AND HOLDS RHLA FROM MOVING


021612  013777  002026  160010        MOV     @#RELEASE,@RHCS1 ;LOAD RELEASE COMMAND INTO RHCS1

                                        ;SAVE REGISTERS FOR COMPARISON AFTER GO
021620  004037  040542                JSR     R0,@#SAVER      ;SAVE
021624  001622                        RHWC                    ;FROM
021626  003126                        REINTO                  ;TO
021630  000023                        19.                     ;NUMBER OF REGISTERS SAVED
```

```
                                        ;GIVE GO TO RELEASE COMMAND
021632  052777  000001  157770          BIS     #GO,@RHCS1      ;GO TO RELEASE COMMAND
021640  052777  000001  160002          BIS     #DMD,@RHMR      ;SET DMD TO HOLD RHLA

                                        ;CHANGE SAVED REGISTERS TO EXPECTED VALUES
                                        ;AFTER GO HAS BEEN GIVEN TO RELEASE COMMAND
021646  052737  000001  003154          BIS     #DMD,@#REINTO+26;SAVED RHMR
021654  017737  160004  003170          MOV     @RHLA,@#REINTO+42;SAVED RHLA


                                        ;SAVE REGISTERS AGAIN SO THAT COMPARISONS CAN
                                        ;BE DONE
021662  004037  040542                  JSR     R0,@#SAVER      ;SAVE
021666  001622                          RHWC                    ;FROM
021670  002062                          WRFROM                  ;TO
021672  000023                          19.                     ;NUMBER OF REGISTERS SAVED

                                        ;AS UPPER BYTE OF RHAS CAN BE CHANGING IN A DUAL PORT
                                        ;OPERATION THE UPPER BYTE OF RHAS WILL BE SAVED AS IS
                                        ;SO THAT THE COMPARES ARE ONLY VALID FOR THE LOWER BYTE
021674  113737  003153  002107          MOVB    @#REINTO+25,@#WRFROM+25;SAVE UPPER RHAS


                                        ;COMPARE REGISTERS BEFORE RELEASE COMMAND
                                        ;WITH AFTER GO
021702  004037  040736                  JSR     R0,@#COMPAR     ;COMPARE
021706  003126                          REINTO                  ;GOOD BUFFER
021710  002062                          WRFROM                  ;TEST BUFFER
021712  000023                          19.                     ;NUMBER
021714  021722                          1$                      ;RETURN FOR ERROR
021716  021722                          1$                      ;SAME
021720  021742                          2$                      ;RETURN FOR GOOD COMPARISON

021722  013705  044532          1$:     MOV     @#ERWORD,R5     ;GETTING READY TO INDEX
021726  060505                          ADD     R5,R5           ;DOUBLE ERROR WORD
021730  016537  001620  037534          MOV     RHWC-2(R5),@#REGADR ;FAILING REGISTER ADDRESS
021736  104001                          ERROR   1               ;IMPROPER REGISTER CHANGE
                                                                ;AFTER RELEASE COMMAND
                                                                ;WITH GO IS GIVEN
021740  000207                          RTS     PC              ;RETURN TO COMPARISON
021742                          2$:


                                        ;;******************************************************************
                                        ;*TEST 50        MAKE CURRENT CYLINDER = 0
                                        ;;******************************************************************
021742  000004                  TST50:  SCOPE
021744  012706  001000                  MOV     #STACK,SP       ;RESET STACK
021750  004737  040064                  JSR PC,@#CLDISK         ;INIT DRIVE
021754  012777  000001  157666          MOV     #DMD,@RHMR              ;SET DIAGNOSTIC MODE
021762  004037  042400                  JSR     R0,@#MAKECYL            ;SUBROUTINE TO GIVE A SEEK
                                                                        ;COMMAND FOLOWED BY A INIT
                                                                        ;THIS SHUOLD CHANGE RHCC
```

```
 5982  021766  000000                          0                                ;CHANGE RHCC TO 0
 5983
 5984
 5985
 5986
 5987                                  ;;*****************************************************************
 5988                                  ;*TEST 51       LOOK AHEAD REGISTER
 5989
 5990                                  ;*      A SEARCH COMMAND IS GIVEN FOR CYLINDER 0, TRACK 0, SECTOR 21.
 5991                                  ;*      THE LOOK AHEAD REGISTER IS CHECKED AFTER INDEX PULSE
 5992                                  ;*      THE EXTENSION FIELD IS CHECKED IN EACH SECTOR AFTER
 5993                                  ;*      128 BYTES THEN AGAIN AFTER 128 MORE BYTES THEN AGAIN AFTER 256 MORE BYTES
 5994                                  ;*      THE SECTOR COUNT FIELD IS CHECKED AFTER EACH SECTOR
 5995                                  ;*      AT THE END ALL REGISTERS ARE CHECKED
 5996
 5997                                  ;;*****************************************************************
 5998  021770  000004                  TST51:  SCOPE
 5999  021772  012706  001000                  MOV     #STACK,SP       ;RESET STACK
 6000
 6001  021776  012737  000051  004172          MOV     #TTNO,@#TSTNM   ;THIS SAVES TEST NUMBER
 6002
 6003  022004  004737  040064                  JSR     PC,@#CLDISK     ;INIT AND SET UP GENERAL REGISTERS
 6004
 6005                                          ;THESE ARE REGULAR SET UPS FOR SEARCH COMMAND
 6006  022010  012777  000025  157616          MOV     #21.,@RHDST     ;DESIRED SECTOR/TRACK REGISTER
 6007                                                                  ;TRACK 0 SECTOR 21
 6008  022016  005077  157620                  CLR     @RHCA           ;DESIRED CYLINDER =0
 6009  022022  012777  010000  157610          MOV     #FMT22,@RHOF    ;FORMAT BIT=1 (16 BITS PER WORD)
 6010  022030  013711  002030                  MOV     @#SERCH,@R1     ;FILL SEARCH COMMAND IN RHCS1
 6011
 6012                                          ;NOW SAVE REGISTERS STARTING FROM RHWC IN WRITE FROM BUFFER
 6013  022034  004037  040542                  JSR     R0,@#SAVER      ;SAVE REGISTERS FOR COMPARISON
 6014                                                                  ;AT THE END OF THE SEARCH
 6015  022040  001622                          RHWC                    ;START SAVING FROM RHWC
 6016  022042  003126                          REINTO                  ;SAVE INTO REINTO
 6017  022044  000023                          19.                     ;NUMBER OF REGISTERS SAVED
 6018
 6019
 6020
 6021  022046  004767  016046                  JSR     PC,CHECKT               ;CHECK DVA, RDY, DPR, DRY
 6022
 6023
 6024                                          ;NOW THE DIAGNOSTIC MODE BIT WILL BE SET
 6025                                          ;AND THE SEARCH OPERATION STARTED
 6026
 6027  022052  005037  001200                  CLR     @#$TMP1         ;THIS WILL HAVE THE EXPECTED
 6028                                                                  ;VALUE OF RHLA REGISTER
 6029
 6030  022056  013700  001650                  MOV     @#RHMR,R0       ;NOW R0 HAS MAINTENANCE REG. ADDR.
 6031  022062  017703  157546                  MOV     @RHDST,R3       ;GET DESIRED SECTOR/TRACK REG.
 6032  022066  042703  177400                  BIC     #177400,R3      ;GET SECTOR ONLY
 6033  022072  010337  050424                  MOV     R3,@#SECTR      ;DUPLICATE SECTOR
 6034  022076  012710  000001                  MOV     #DMD,@R0        ;S
 6035  022102  052777  000001  157520          BIS     #GO,@RHCS1      ;GO
 6036  022110  052710  000010                  BIS     #MSTCK,@R0      ;SET SECTOR CLOCK
 6037  022114  042710  000010                  BIC     #MSTCK,@R0      ;CLEAR SECTOR CLOCK
```

```
6038  022120  000240                        NOP                     ;ALLOW TIME BETWEEN SECTOR CLOCKS
6039  022122  052710  000010                BIS     #MSTCK,@R0      ;SET SECTOR CLOCK
6040  022126  042710  000010                BIC     #MSTCK,@R0      ;CLEAR SECTOR CLOCK
6041  022132  000240                        NOP                     ;ALLOW TIME BETWEEN SECTOR CLOCKS
6042  022134  052710  000014                BIS     #MINX!MSTCK,@R0 ;SET INDEX AND SECTOR CLOCK
6043  022140  012710  000001                MOV     #DMD,@R0        ;RESET INDEX AND SECTOR CLOCK
6044  022144  005703                        TST     R3              ;IF SECTOR REQUIRED JUMP OUT
6045  022146  001555                        BEQ     11$             ;BRANCH OF SECTOR ZERO REQUIRED
6046
6047                                        ;AFTER THE INDEX PULSE RHLA WILL BE CHECKED TO BE ZERO
6048                                        ;AND $TMP4 WILL BE SET UP TO COUNT BYTES
6049  022150  012737  001140  001206  1$:   MOV     #608.,@#$TMP4   ;THERE ARE 608 BYTES PER SECTOR
6050  022156  017737  157502  001126        MOV     @RHLA,@#$BDDAT  ;SAVE RHLA
6051  022164  017737  157444  001704        MOV     @RHDST,@#DST    ;SAVE DESIRED SECTOR TRACK
6052  022172  023737  001200  001126        CMP     @#$TMP1,@#$BDDAT ;RHLA SHOULD BE HAVE EXTENSION
6053                                                                ;FIELD EQUAL TO ZERO
6054  022200  001414                        BEQ     2$              ;BRANCH IF GOOD
6055  022202  013737  050424  001202        MOV     @#SECTR,@#$TMP2 ;GET SECTOR SOUGHT
6056  022210  160337  001202                SUB     R3,@#$TMP2      ;$TMP2 NOW HAS PRESENT SECTOR
6057  022214  012746  001140                MOV     #608.,-(SP)     ;NUMBER OF BYTES PER SECTOR
6058  022220  163716  001206                SUB     @#$TMP4,(SP)    ;(SP)HAS PRESENT BYTE NUMBER
6059  022224  012637  001204                MOV     (SP)+,@#$TMP3   ;PRESENT BYTE NUMBER
6060  022230  104024                        ERROR   24              ;LOOK AHEAD REGISTER AT THE BEGINING OF A
6061                                                                ;SECTOR IS IN ERROR
6062
6063
6064                                        ;NOW THE 304 WORDS WILL START
6065                                        ;FOR FIRST BYTE CLOCK WILL BE INDEPENDENT OF
6066                                        ;SECTOR CLOCK THEN IT WILL COINCIDE FOREVER TILL
6067                                        ;THE BEGINNING OF NEXT SECTOR
6068
6069                                        ;ONE WORD ONLY THAT IS TWO BYTES
6070
6071  022232  012702  000010        2$:     MOV     #8.,R2          ;BYTE
6072  022236  012705  000002                MOV     #2,R5           ;BYTES PER WORD
6073  022242  000404                        BR      4$
6074  022244  052710  000012        3$:     BIS     #MSTCK!MCLK,@R0 ;SET SECTOR AND CLOCK
6075  022250  042710  000012                BIC     #MSTCK!MCLK,@R0 ;CLEAR SECTOR AND CLOCK
6076  022254  052710  000002        4$:     BIS     #MCLK,@R0       ;SET CLOCK
6077  022260  042710  000002                BIC     #MCLK,@R0       ;CLEAR CLOCK
6078  022264  005302                        DEC     R2              ;BYTE COUNTER
6079  022266  001372                        BNE     4$              ;BRANCH IF BYTE NOT COMPLETE
6080  022270  005337  001206                DEC     @#$TMP4         ;BYTE COUNT DOWN
6081  022274  012702  000007                MOV     #7,R2           ;SETUP FOR SECOND BYTE
6082  022300  005305                        DEC     R5              ;IS WORD COMPLETE?
6083  022302  001360                        BNE     3$              ;BRENCH IF NOT COMPLETE
6084                                                                ;TO GIVE SECTOR CLOCK AND CLOCK
6085
6086                                        ;NOW 303 WORDS ARE LEFT ALL ARE IDENTICAL
6087                                        ;THAT IS 606 IDENTICAL BYTES WILL BE GIVEN
6088                                        ;RHLA WILL BE CHECKED STAR TO COUNT AFTER
6089                                        ;BEGINNING OF SECTOR PULSE
6090                                        ;AFTER 128 BYTES (2 BYTES ARE ALREADY GIVEN)
6091                                        ;SO 127 MORE
6092                                        ;THEN RHLA WILL BE CHECKED AFTER 128 MORE BYTES
6093                                        ;THEN RHLA WILL BE CHECKED AFTER 256 MORE BYTES
```

```
                                        ;THEN THE TOTAL OF 608 BYTES WILL BE COMPLETED
                                        ;AND RHLA WILL BE MADE READY FOR NEXT SECTOR
                                        ;AND RHLA WILL BE CHECKED

022304  012705  000100                  MOV     #64.,R5         ;R5 WILL KEEP TRACK WHEN
                                                                ;EXTENSION FIELD IS TO BE CHECKED
022310  012701  000177                  MOV     #127.,R1        ;FIRST TIME CHECK EXTENSION FIELD
                                                                ;AFTER 127 MORE BYTES
022314  012702  000007          5$:     MOV     #7,R2           ;CLOCKS PER BYTE COUNTER
022320  052710  000012                  BIS     #MSTCK!MCLK,@R0 ;SET SECTOR CLOCK AND CLOCK
022324  042710  000012                  BIC     #MSTCK!MCLK,@R0 ;CLEAR SECTOR CLOCK AND CLOCK
022330  052710  000002          6$:     BIS     #MCLK,@R0       ;SET CLOCK
022334  042710  000002                  BIC     #MCLK,@R0       ;RESET CLOCK
022340  005302                          DEC     R2              ;COUNT DOWN CLOCKS PER BYTE
022342  001372                          BNE     6$              ;BRANCH IF BYTE NOT COMPLETE
022344  005337  001206                  DEC     @#$TMP4         ;COUNT DOWN BYTES
022350  001436                          BEQ     10$             ;BRANCHOUT IF 608 BYTES DONE
022352  005301                          DEC     R1              ;COUNT DOWN NUMBER OF BYTES
                                                                ;TO CHECK EXTENSION FIELD
022354  001357                          BNE     5$              ;BRANCH IF EXTENSION FIELD NOT
                                                                ;TO BE CHECKED YET

                                        ;NOW THE EXTENSION FIELD OF THE LOOK AHEAD REGISTER
                                        ;WILL BE CHECKED

022356  062737  000020  001200          ADD     #20,@#$TMP1     ;GET TO THE NEXT EXTENSION
022364  017737  157274  001126          MOV     @RHLA,@#$BDDAT  ;GET RHLA FOR COMPARISON

022372  017737  157236  001704          MOV     @RHDST,@#DST    ;SAVE DESIRED SECTOR TRACK
022400  023737  001200  001126          CMP     @#$TMP1,@#$BDDAT ;CHECK VALUE OF RHLA
022406  001414                          BEQ     7$              ;BRANCH IF GOOD
022410  013737  050424  001202          MOV     @#SECTR,@#$TMP2 ;GET SECTOR SOUGHT
022416  160337  001202                  SUB     R3,@#$TMP2      ;$TMP2 NOW HAS PRESENT SECTOR
022422  012746  001140                  MOV     #608.,-(SP)     ;NUMBER OF BYTES PER SECTOR
022426  163716  001206                  SUB     @#$TMP4,(SP)    ;(SP) HAS PRESENT BYTE NUMBER
022432  012637  001204                  MOV     (SP)+,@#$TMP3   ;PRESENT BYTE NUMBER
022436  104025                          ERROR   25              ;LOOK AHEAD ERROR IN THE MIDDLE
                                                                ;OF A SECTOR IS IN ERROR

022440  060505                  7$:     ADD     R5,R5           ;GET NEXT STEP TO CHECK EXTENSION FIELD
022442  010501                          MOV     R5,R1           ;PUT IN COUNTER
022444  000723                          BR      5$              ;BRANCH BACK SECTOR
                                                                ;IS NOT COMPLETE
022446  062737  000020  001200  10$:     ADD     #20,@#$TMP1
022454  052710  000010                  BIS     #MSTCK,@R0      ;THESE TWO INSTRUCTIONS GIVE
022460  042710  000010                  BIC     #MSTCK,@R0      ;ONE SECTOR CLOCK EXTRA
022464  000240                          NOP                     ;ALLOW TIME BETWEEN SECTOR CLOCK
022466  052710  000010                  BIS     #MSTCK,@R0      ;THESE TWO INSTRUCTIONS GIVE
022472  042710  000010                  BIC     #MSTCK,@R0      ;ONE SECTOR CLOCK EARLY
                                                                ;BEFORE THE NEXT SECTOR
022476  005303                          DEC     R3              ;IS REQUIRED NO OF SECTORS COMPLETE
022500  001223                          BNE     1$              ;BRANCH IF NOT

                                        ;NOW THE REQUIRED SECTOR IS REACHED
                                        ;ONE SECTOR CLOCK WILL BE GIVEN TO GET SECTOR PULSE
                                        ;DOWN AND HENCE ATA UP
```

```
                                11$:    MOV     #8.,R2          ;8 CLOCKS
                                12$:    BIS     #MCLK,@R0       ;SET CLOCK
                                        BIC     #MCLK,@R0       ;CLEAR CLOCKS
                                        DEC     R2              ;COUND DOWN
                                        BNE     12$             ;BRANCH IF 8 NOT DONE
                                        BIS     #MSTCK!MCLK,@R0 ;SET SECTOR AND CLOCK
                                        BIC     #MSTCK!MCLK,@R0 ;CLEAR SECTOR AND CLOCK

                                ;NOW ALL REGISTERS WILL BE COMPARED
                                ;SO FILL EXPECTED VALUE INTO SAVED LOCATIONS

                                        BIS     #SC,@#REINTO+6  ;INCLUDE SC IN SAVED RHCS1
                                        BIS     @#ATTENT,@#REINTO+24 ;FILL APPROPIATE ATTENTION
                                                                ;IN SAVED RHAS
                                        BIS     #DMD,@#REINTO+26 ;SET DMD IN RHMR SAVED
                                        BIS     #ATA,@#REINTO+30 ;SET ATA IN RHDS1 SAVED
                                        MOV     @#$TMP1,@#REINTO+42 ;MOVE EXPECTED VALUE
                                                                ;INTO RHLA SAVED

                                ;AFTER SEARCH COMMAND
                                ;SAVE REGISTERS AGAIN SO THAT COMPARES CAN BE DONE

                                        JSR     R0,@#SAVER      ;SAVE
                                        RHWC                    ;FROM
                                        WRFROM                  ;TO
                                        19.                     ;NUMBER

                                ;AS UPPER BYTE OF RHAS CAN BE CHANGING IN A DUAL PORT
                                ;OPERATION THE UPPER BYTE OF RHAS WILL BE SAVED AS IS
                                ;SO THAT THE COMPARES ARE ONLY VALID FOR THE LOWER BYTE
                                        MOVB    @#REINTO+25,@#WRFROM+25;SAVE UPPER RHAS


                                ;COMPARE REGISTERS BEFORE SEARCH WITH AFTER

                                        JSR     R0,@#COMPAR     ;COMPAR
                                        REINTO                  ;GO BUFFER

                                        WRFROM                  ;TEST BUFFER
                                        18.                     ;NUMBER
                                        13$                     ;RETURN FOR ERROR
                                        13$                     ;SAME
                                        14$                     ;RETURN FOR GOOD COMPARISON
                                13$:    MOV     @#ERWORD,R5     ;GETTING READY TO INDEX
                                        ADD     R5,R5           ;DOUBLE ERROR WORD
                                        MOV     RHWC-2(R5),@#REGADR ;FAILING REG. ADDRESS
                                        ERROR   1               ;CONTENTS OF REGISTER
                                        RTS     PC              ;CHANGED AT END OF
                                14$:                            ;SEARCH
                                ;;**************************************************************
                                ;*TEST 52        MAKE CURRENT CYLINDER = 0
                                ;;**************************************************************
                                TST52:  SCOPE
                                        MOV     #STACK,SP       ;RESET STACK
                                        JSR PC,@#CLDISK         ;INIT DRIVE
```

```
6206  022662  012777  000001  156760          MOV     #DMD,@RHMR              ;SET DIAGNOSTIC MODE
6207  022670  004037  042400                  JSR     R0,@#MAKECYL            ;SUBROUTINE TO GIVE A SEEK
6208                                                                          ;COMMAND FOLOWED BY A INIT
6209                                                                          ;THIS SHUOLD CHANGE RHCC
6210  022674  000000                          0                               ;CHANGE RHCC TO 0
6211
6212
6213
6214
6215
6216                                  ;;**************************************************************
6217                                  ;*TEST 53       WRITE HEADER AND DATA 1
6218
6219                                  ;*      WRITE CYLINDER 0, FORMAT 16 BIT PER WORD
6220                                  ;*      TRACK 0, SECTOR 0, KEYS 0, NUMBER OF WORDS 256 OF 0'S
6221                                  ;*      AS EVERYTHING IS ZERO THIS PROVES VERY LITTLE
6222                                  ;*      ANY ERROR LOGIC INDICATIONS ARE NOT CONCLUSIVE ON FIRST PASS
6223                                  ;*      BECAUSE ERROR LOGIC HAS NOT BEEN CHECKED YET (ON FIRST PASS)
6224
6225                                  ;;**************************************************************
6226  022676  000004                  TST53:  SCOPE
6227  022700  012706  001000                  MOV     #STACK,SP               ;RESET STACK
6228
6229
6230  022704  012737  000053  004172          MOV     #TTNO,@#TSTNM           ;THIS SAVES TEST NUMBER
6231
6232  022712  012700  046232                  MOV     #SECGAP,R0              ;POINTER
6233  022716  012701  000460                  MOV     #304.,R1                ;COUNTER
6234  022722  012720  177777          1$:     MOV     #-1,(R0)+               ;CLEAR DISK AREA TO ALL ONES.
6235  022726  005301                          DEC     R1                      ;
6236  022730  001374                          BNE     1$                      ;
6237  022732  004767  015126                  JSR     PC,CLDISK                       ;THIS IS USED TO SET GENERAL
6238                                                                          ;REGISTERS
6239
6240                                  ;THESE ARE TO SET UP FOR DISKLESS USE ONLY
6241
6242  022736  012737  010000  047516          MOV     #FMT22,@#WCYL           ;FORMAT 22=16 BITWORDS AND
6243                                                                          ;CYLINDER 0
6244  022744  005037  047520                  CLR     @#WSECTR                ;TRACK=0, SECTOR=0
6245  022750  005037  047522                  CLR     @#WKEY1                 ;KEY1=0
6246  022754  005037  047524                  CLR     @#WKEY2                 ;KEY2=0
6247  022760  012737  000400  047556          MOV     #256.,@#FNWORD          ;256 DATAWORDS
6248  022766  004537  041242                  JSR     R5,@#CRC                ;GO TO CALCULATE CRC
6249  022772  047516                          WCYL
6250  022774  047526                          GCRC
6251
6252                                  ;THESE ARE REGULAR SETUPS
6253
6254  022776  012777  177374  156616          MOV     #-260.,@RHWC            ;256 DATA WORDS 4 HEADER WORDS
6255  023004  012700  002062                  MOV     #WRFROM,R0              ;FROM BUFFER "WRFROM"
6256  023010  010077  156610                  MOV     R0,@RHBA                ;IN BUS ADDRESS
6257  023014  012705  000403                  MOV     #259.,R5                ;COUNTER
6258  023020  012720  010000                  MOV     #FMT22,(R0)+            ;FORMAT =16 BIT WORD
6259                                                                          ;CYLINDER=0
6260  023024  005020                  2$:     CLR     (R0)+                   ;SECTOR=0, TRACK=0,KEYS=0, ALL DATA=0
6261  023026  005305                          DEC     R5                      ;COUNT
```

```
6262  023030  001375                        BNE     2$              ;BRANCH IF ALL 259 NOT COMPLETE
6263  023032  005077  156576                CLR     @RHDST          ;TRACK=0, SECTOR=0
6264
6265
6266
6267  023036  004767  015056                JSR     PC,CHECKT               ;CHECK DVA, RDY, DFR, DRY
6268
6269  023042  013711  002040                MOV     @#WRIFOR,@R1    ;GET READY FOR WRITE HEADER
6270                                                                ;AND DATA WITH 62 IN RHCS1
6271  023046  005037  001772                CLR     @#ERFLG$        ;CLEAR ERROR FLAG
6272  023052  012777  010000  156560        MOV     #FMT22,@RHOF    ;FORMAT BIT=1 16 BIT WORDS
6273  023060  005077  156556                CLR     @RHCA           ;CYLINDER 0
6274  023064  004737  047372                JSR     PC,@#COMWHD     ;WRITE HEADER AND DATA
6275
6276                                        ;IF THE PROGRAM COMES BACK HERE WITHOUT ERROR
6277                                        ;PRINT OUTS FROM THE "COMWHD" ROUTINE THAT MEANS
6278                                        ;ALL HEADER ON DISK IS GOOD IE. ONLY DATA IS
6279                                        ;TO BE CHECKED TO SEE IF IT IS ZERO
6280                                        ;AND WITE DATA GAP AND TOLERANCE GAP TO SEE IF THEY
6281                                        ;ARE ALL ZEROS, ONLY ECC1 AND ECC2 MAY NOT BE 0
6282
6283  023070  005737  001772                TST     @#ERFLG$                ;HAS ANY ERRORS OCCURED?
6284
6285  023074  001031                        BNE     TST54   ;;BRANCH IF YES
6286
6287  023076  004737  040302                JSR     PC,@#CHECKE     ;CHECK DVA,RDY,DRY,DPR
6288  023102  005037  047330                CLR     @#WECC1         ;CLEAR ECC
6289  023106  005037  047332                CLR     @#WECC2
6290                              ;REINTO BUFFER IS FILLED WITH EXPECTED DATA
6291  023112  004037  040002                JSR     R0,@#CLAREA     ;CLEAR "REINTO"
6292  023116  003126                        REINTO                  ;FROM
6293  023120  004166                        REINTO+<272.*2>                 ;TO
6294  023122  000000                        .WORD   0               ;FILL WITH ZEROS
6295
6296
6297  023124  005037  001772                CLR     @#ERFLG$        ;CLEAR ERROR FLAG
6298
6299
6300                                    ;COMPARE "REINTO" WITH "DISK" BUFFER
6301  023130  004037  040736                JSR     R0,@#COMPAR     ;CHECK
6302  023134  003126                        REINTO                  ;GOOD BUFFER
6303  023136  046330                        DISK                    ;TEST BUFFER
6304  023140  000421                        273.                    ;NUMBER OF WORDS CHECKED
6305  023142  023150                        3$                      ;RETURN POINT FOR ERROR HEADER
6306  023144  023154                        4$                      ;RETURN POINT FOR ERROR DATA
6307
6308  023146  023160                        TST54                   ;RETURN FOR GOOD COMPARISON
6309
6310  023150  104007              3$:       ERROR   7               ;READ ERROR10 NEXT
6311  023152  000207                        RTS     PC              ;RETURN TO "COMPAR"
6312  023154  104010              4$:       ERROR   10              ;WORD NOS 1 TO 256 ARE
6313                                                                ;DATA WORDS
6314                                                                ;257 AND 258 ARE ECC
6315                                                                ;ZEROED OUT
6316                                                                ;259 TO 273 TOLERANCE GAP
6317  023156  000207                        RTS     PC
```

```
                                ;;*****************************************************************
                                ;*TEST 54       WRITE HEADER AND DATA 2
                                ;
                                ;*      WRITE CYLINDER0, FORMAT 16 BITS PER WORD
                                ;*      TRACK 0, SECTOR 1, KEYS 0, NUMBER OF WORDS 256
                                ;*      OF ALL ONES.
                                ;*      ANY ERROR LOGIC INDICATIONS ARE NOT CONCLUSIVE ON FIRST PASS
                                ;*      BECAUSE ERROR LOGIC HAS NOT BEEN CHECKED YET (ON FIRST PASS)
                                ;
                                ;;*****************************************************************
023160  000004                  TST54:  SCOPE
023162  012706  001000                  MOV     #STACK,SP       ;RESET STACK
                                ;
023166  012737  000054  004172          MOV     #TTNO,@#TSTNM   ;THIS SAVES TEST NUMBER
                                ;
023174  012700  046232                  MOV     #SECGAP,R0      ;POINTER
023200  012701  000460                  MOV     #304.,R1        ;COUNTER
023204  005020                  1$:     CLR     (R0)+           ;CLEAR SIMULATED DISK AREA
023206  005301                          DEC     R1
023210  001375                          BNE     1$
023212  004737  040064                  JSR     PC,@#CLDISK                     ;THIS IS USED TO SET GENERAL REGISTERS
                                ;
                                ;THESE ARE TO BE SETUP FOR DISKLESS USE ONLY
                                ;
023216  012737  010000  047516          MOV     #FMT22,@#WCYL   ;FORMAT22=16BIT WORDS AND
                                                                ;CYLINDER 0
023224  012737  000001  047520          MOV     #1,@#WSECTR     ;TRACK=0, SECTOR=1
023232  005037  047522                  CLR     @#WKEY1         ;KEY1=0
023236  005037  047524                  CLR     @#WKEY2         ;KEY2=0
023242  012737  000400  047556          MOV     #256.,@#FNWORD  ;256 DATA WORDS
023250  004537  041242                  JSR     R5,@#CRC        ;GO TO CALCULATE CRC
023254  047516                          WCYL
023256  047526                          GCRC
                                ;
                                ;THESE ARE REGULAR SETUPS
                                ;
023260  012777  177374  156334          MOV     #-260.,@RHWC    ;256 DATA WORDS 4 HEADER WORDS
023266  012700  002062                  MOV     #WRFROM,R0      ;THESE TWO INSTRUCTIONS GETS
023272  010077  156326                  MOV     R0,@RHBA        ;ADDR. OF WRFROM INTO R0 AND
                                                                ;BUS ADDRESS REGISTER
023276  012720  010000                  MOV     #FMT22,(R0)+    ;FORMAT=16 BIT WORDS
                                                                ;CYLINDER=0
023302  012720  000001          2$:     MOV     #1,(R0)+        ;TRACK=0, SECTOR=1, KEYS=0
023306  005020                          CLR     (R0)+           ;KEY1=0
023310  005020                          CLR     (R0)+           ;KEY2=0
023312  012705  000400                  MOV     #256.,R5        ;COUNTER
023316  012720  177777          3$:     MOV     #-1,(R0)+       ;MOVE ALL ONES FOR DATA
023322  005305                          DEC     R5
023324  001374                          BNE     3$              ;BRANCH IF DATA NOT COMPLETE
023326  012777  000001  156300          MOV     #1,@RHDST       ;TRACK=0 SECTOR=1
```

```
6374  023334  004767  014560                  JSR     PC,CHECKT               ;CHECK DVA, RDY, DPR, DRY
6375
6376
6377  023340  013711  002040                  MOV     @#WRIFOR,@R1    ;GET READY FOR WRITE HEADER AND
6378                                                                  ;DATA WITH 62 IN RHCS1
6379  023344  005037  001772                  CLR     @#ERFLG$                ;CLEAR ERROR FLAG
6380  023350  012777  010000  156262          MOV     #FMT22,@RHOF    ;FORMAT BIT=1 (16 BIT WORDS)
6381  023356  005077  156260                  CLR     @RHCA                   ;CYLINDER =0
6382  023362  004737  047372                  JSR     PC,@#COMWHD     ;WRITE HEADER AND DATA
6383
6384                                          ;IF THE PROGRAM COMES BACK HERE WITHOUT ERROR PRINTOUTS
6385                                          ;FROM THE "COMWHD" ROUTINE THAT MEANS ALL HEADER ON DISK
6386                                          ;IS GOOD IE. ONLY DATA IS TO BE CHECKED TO SEE IF THEY ARE
6387                                          ;ALL ONES AND WRITE DATA GAP AND TOLERANCE GAP TO SEE IF
6388                                          ;THEY ARE ALL ZEROS, ONLY ECC1 AND ECC2 ARE NOT CHECKED
6389
6390  023366  005737  001772                  TST     @#ERFLG$                ;HAS ANY ERRORS OCCURED?
6391
6392  023372  001036                          BNE     TST55   ;;BRANCH IF YES
6393
6394  023374  004737  040302                  JSR     PC,@#CHECKE     ;CHECK DVA,RDY,DRY,DPR
6395  023400  005037  047330                  CLR     @#WECC1         ;CLEAR ECC
6396  023404  005037  047332                  CLR     @#WECC2         ;CLEAR ECC
6397                                  ;FILL "REINTO" BUFFER WITH EXPECTED DATA
6398  023410  004037  040002                  JSR     R0,@#CLAREA     ;FILL REINTO BUFFER
6399  023414  003126                                  REINTO          ;FROM
6400  023416  004124                                  REINTO+<255.*2>         ;TO
6401  023420  177777                          .WORD   -1              ;DATA
6402  023422  004037  040002                  JSR     R0,@#CLAREA     ;FILL REST
6403  023426  004126                          REINTO+<256.*2>         ;FROM
6404  023430  004166                          REINTO+<272.*2>         ;TO
6405  023432  000000                          0                       ;DATA
6406
6407
6408  023434  005037  001772                  CLR     @#ERFLG$                ;CLEAR ERROR FLAG
6409
6410
6411                                  ;NOW COMPARE "DISK" BUFFER WITH "REINTO"
6412  023440  004037  040736                  JSR     R0,@#COMPAR     ;CHECK
6413  023444  003126                          REINTO                  ;GOOD BUFFER
6414  023446  046330                          DISK                    ;TEST BUFFER
6415  023450  000421                          273.                    ;NUMBER OF WORDS CHECKED
6416  023452  023460                          4$                      ;RETURN POINT FOR ERROR HEADER
6417  023454  023464                          5$                      ;RETURN POINT FOR ERROR DATA
6418
6419  023456  023470                          TST55                   ;RETURN FOR GOOD COMPARISON
6420
6421  023460  104007                  4$:     ERROR   7               ;READ ERROR 10 NEXT
6422  023462  000207                          RTS     PC              ;RETURN TO COMPARE
6423  023464  104010                  5$:     ERROR   10              ;WORD NOS 1 TO 256 ARE
6424                                                                  ;DATA WORDS
6425                                                                  ;WORD NOS 257 AND 258
6426                                                                  ;ARE ECC WHICH HAVE BEEN
6427                                                                  ;ZEROED
6428                                                                  ;WORD NOS 259
6429                                                                  ;IS DATA GAP
```

```
                                                        ;WORD NOS 260 TO 273
                                                        ;ARE TOLERANCE GAP
023466  000207                  RTS     PC              ;RETURN TO COMPARE


                        ;;*****************************************************************
                        ;*TEST 55       WRITE HEADER AND DATA 3
                        ;*      WRITE CYLINDER 0 FORMAT 16 BITS PER WORD
                        ;*      TRACK 1, SECTOR 1, KEY 0, NUMBER OF WORDS 256
                        ;*      ALTERNATE ONES AND ZEROS (052525)
                        ;*      ANY ERROR LOGIC INDICATIONS ARE NOT CONCLUSIVE ON FIRST PASS
                        ;*      BECAUSE ERROR LOGIC HAS NOT BEEN CHECKED YET (ON FIRST PASS)
                        ;;*****************************************************************
023470  000004         TST55:  SCOPE
023472  012706  001000          MOV     #STACK,SP       ;RESET STACK

023476  012737  000055  004172  MOV     #TTNO,@#TSTNM   ;THIS SAVES TEST NUMBER

023504  012700  046232          MOV     #SECGAP,R0      ;POINTER
023510  012701  000460          MOV     #304.,R1        ;COUNTER
023514  012720  000377  1$:     MOV     #377,(R0)+      ;CLEAR SIMULATED DISK AREA WITH 377
023520  005301                  DEC     R1
023522  001374                  BNE     1$
023524  004767  014334          JSR     PC,CLDISK                       ;THIS IS USED TO SET GENERAL
                                                        ;REGISTERS

                        ;THESE ARE TO BE SETUP FOR DISKLESS USE ONLY

023530  012737  010000  047516  MOV     #FMT22,@#WCYL   ;FORMAT 22=16 BIT WORDS AND
                                                        ;CYLINDER 0
023536  012737  000401  047520  MOV     #401,@#WSECTR   ;TRACK=1, SECTOR=1
023544  005037  047522          CLR     @#WKEY1         ;KEY1=0
023550  005037  047524          CLR     @#WKEY2         ;KEY2=0
023554  012737  000400  047556  MOV     #256.,@#FNWORD  ;256 DATA WORDS
023562  004537  041242          JSR     R5,@#CRC        ;GO TO CALCULATE CRC
023566  047516                  WCYL
023570  047526                  GCRC

                        ;THESE ARE REGULAR SETUPS

023572  012777  177374  156022  MOV     #-260.,@RHWC    ;256 DATA WORDS 4 HEADER WORDS
023600  012700  002062          MOV     #WRFROM,R0      ;THESE TWO INSTRUCTIONS GETS
                                                        ;ADDR. OF WRFROM INTO R0
023604  010077  156014          MOV     R0,@RHBA        ;AND BUS ADDRESS REGISTER

023610  012720  010000          MOV     #FMT22,(R0)+    ;FORMAT=16 BIT WORDS
                                                        ;CYLINDER=0
023614  012720  000401  2$:     MOV     #401,(R0)+      ;TRACK=1, SECTOR=1, KEYS=0
023620  005020                  CLR     (R0)+           ;KEY1=0
023622  005020                  CLR     (R0)+           ;KEY2=0
023624  012705  000400          MOV     #256.,R5        ;COUNTER
023630  012720  052525  3$:     MOV     #052525,(R0)+   ;MOVE ALTERNAT ONES FOR DATA
```

```
6486  023634  005305                        DEC     R5              ;COUNT
6487  023636  001374                        BNE     3$              ;BRANCH IF DATA NOT COMPLETE
6488  023640  012777  000401  155766        MOV     #401,@RHDST     ;TRACK=1 SECTOR=1
6489
6490
6491
6492  023646  004767  014246                JSR     PC,CHECKT               ;CHECK DVA, RDY, DPR, DRY
6493
6494  023652  013711  002040                MOV     @#WRIFOR,@R1    ;GET READY FOR WRITE HEADER
6495                                                                ;AND DATA WITH 62 IN RHCS1
6496  023656  005037  001772                CLR     @#ERFLG$        ;CLEAR ERROR FLAG
6497  023662  012777  010000  155750        MOV     #FMT22,@RHOF    ;FORMAT BIT=1(16 BIT WORDS
6498  023670  005077  155746                CLR     @RHCA           ;CYLINDER=0
6499  023674  004737  047372                JSR     PC,@#COMWHD     ;WRITE HEADER AND DATA
6500
6501                                        ;IF THE PROGRAM COMES BACK HERE WITHOUT ERROR PRINTOUTS
6502                                        ;FROM THE "COMWHD" ROUTINE THAT MEANS ALL HEADER ON
6503                                        ;DISK IS GOOD IE.  ONLY DATA IS TO BE CHECKED TO SEE
6504                                        ;IF THEY ARE ALL 052525 AND WRITE DATA GAP AND
6505                                        ;TOLERANCE GAP TO SEE IF THEY ARE ALL ZEROS
6506                                        ;ONLY ECC1 AND ECC2 ARE NOT CHECKED
6507
6508  023700  005737  001772                TST     @#ERFLG$                ;HAS ANY ERRORS OCCURED?
6509
6510  023704  001036                        BNE     TST56   ;;BRANCH IF YES
6511
6512  023706  004737  040302                JSR     PC,@#CHECKE     ;CHECK DVA,RDY,DRY,DPR
6513  023712  005037  047330                CLR     @#WECC1         ;CLEAR ECC
6514  023716  005037  047332                CLR     @#WECC2         ;CLEAR ECC
6515                              ;FILL "REINTO" BUFFER WITH EXPECTED DATA
6516  023722  004037  040002                JSR     R0,@#CLAREA     ;FILL REINTO BUFFER
6517  023726  003126                        REINTO                  ;FROM
6518  023730  004124                        REINTO+<255.*2>         ;TO
6519  023732  052525                        .WORD   52525           ;DATA
6520  023734  004037  040002                JSR     R0,@#CLAREA     ;FILL REST
6521  023740  004126                                REINTO+<256.*2>         ;FROM
6522  023742  004166                        REINTO+<272.*2>         ;TO
6523  023744  000000                        .WORD   0               ;DATA
6524
6525
6526  023746  005037  001772                CLR     @#ERFLG$        ;CLEAR ERROR FLAG
6527
6528
6529                              ;NOW COMPARE "DISK" BUFFER WITH "REINTO"
6530  023752  004037  040736                JSR     R0,@#COMPAR     ;CHECK
6531  023756  003126                        REINTO                  ;GOOD BUFFER
6532  023760  046330                        DISK                    ;TEST BUFFER
6533  023762  000421                        273.                    ;NUMBER OF WORDS CHECKED
6534  023764  023772                        4$                      ;RETURN POINT FOR ERROR HEADER
6535  023766  023776                        5$                      ;RETURN POINT FOR ERROR DATA
6536
6537  023770  024002                        TST56                   ;RETURN FOR GOOD COMPARISON
6538
6539  023772  104007              4$:       ERROR   7               ;READ ERROR 10 NEXT
6540  023774  000207                        RTS     PC              ;RETURN TO COMPARE
6541  023776  104010              5$:       ERROR   10              ;WORD NOS 1 TO 256 ARE
```

```
                                                        ;DATA WORDS
                                                        ;WORD NOS 257 AND 258
                                                        ;ARE ECC WHICH HAVE BEEN
                                                        ;ZEROED
                                                        ;WORD NOS 259
                                                        ;IS DATA GAP
                                                        ;WORD NOS 260 TO 273
                                                        ;ARE TOLERANCE GAP
024000  000207                  RTS     PC              ;RETURN TO COMPARE



                        ;;*****************************************************************
                        ;*TEST 56        PROGRAM ERROR RHCS2 #10

                        ;*      WRITE CYLINDER 0, FORMAT 16 BIT PER WORD
                        ;*      TRACK 0, SECTOR 0, KEYS 0, NUMBER OF WORDS 256 OF 0'S
                        ;*      WHILE GO BIT IS SET ANOTHER GO IS GIVEN THIS SHOULD SET
                        ;*      PROGRAM ERROR

                        ;;*****************************************************************
024002  000004          TST56:  SCOPE
024004  012706  001000          MOV     #STACK,SP       ;RESET STACK

024010  012737  000056  004172  MOV     #TTNO,@#TSTNM   ;THIS SAVES TEST NUMBER

024016  012700  046232          MOV     #SECGAP,R0      ;POINTER
024022  012701  000460          MOV     #304.,R1        ;COUNTER
024026  012720  177777  1$:     MOV     #-1,(R0)+       ;CLEAR DISK AREA TO ALL ONES.
024032  005301                  DEC     R1              ;
024034  001374                  BNE     1$              ;
024036  004767  014022          JSR     PC,CLDISK                       ;THIS IS USED TO SET GENERAL
                                                        ;REGISTERS

                        ;THESE ARE TO SET UP FOR DISKLESS USE ONLY

024042  012737  010000  047516  MOV     #FMT22,@#WCYL   ;FORMAT 22=16 BITWORDS AND
                                                        ;CYLINDER 0
024050  005037  047520          CLR     @#WSECTR        ;TRACK=0, SECTOR=0
024054  005037  047522          CLR     @#WKEY1         ;KEY1=0
024060  005037  047524          CLR     @#WKEY2         ;KEY2=0
024064  012737  000400  047556  MOV     #256.,@#FNWORD  ;256 DATAWORDS
024072  004537  041242          JSR     R5,@#CRC        ;GO TO CALCULATE CRC
024076  047516                  WCYL
024100  047526                  GCRC

                        ;THESE ARE REGULAR SETUPS

024102  012777  177374  155512  MOV     #-260.,@RHWC    ;256 DATA WORDS 4 HEADER WORDS
024110  012700  002062          MOV     #WRFROM,R0      ;FROM BUFFER "WRFROM"
024114  010077  155504          MOV     R0,@RHBA        ;IN BUS ADDRESS
024120  012705  000403          MOV     #259.,R5        ;COUNTER
024124  012720  010000          MOV     #FMT22,(R0)+    ;FORMAT =16 BIT WORD
```

```
                                                        ;CYLINDER=0
024130  005020                2$:     CLR     (R0)+       ;SECTOR=0, TRACK=0,KEYS=0, ALL DATA=0
024132  005305                        DEC     R5          ;COUNT
024134  001375                        BNE     2$          ;BRANCH IF ALL 259 NOT COMPLETE
024136  005077  155472                CLR     @RHDST      ;TRACK=0, SECTOR=0


024142  004767  013752                JSR     PC,CHECKT           ;CHECK DVA, RDY, DPR, DRY

024146  013711  002040                MOV     @#WRIFOR,@R1    ;GET READY FOR WRITE HEADER
                                                        ;AND DATA WITH 62 IN RHCS1
024152  005037  001772                CLR     @#ERFLG$    ;CLEAR ERROR FLAG
024156  012777  010000  155454        MOV     #FMT22,@RHOF    ;FORMAT BIT=1 16 BIT WORDS
024164  005077  155452                CLR     @RHCA       ;CYLINDER 0
024170  012777  000001  155452        MOV     #DMD,@RHMR  ;SET DIAGNOSTIC MODE
024176  052777  000001  155424        BIS     #GO,@RHCS1  ;GO
024204  000240                        NOP
024206  052777  000001  155414        BIS     #GO,@RHCS1  ;THIS GO SHOULD SET PGE

024214  004737  037470                JSR     PC,@#PUTREG ;SAVE REGISTERS
024220  032737  002000  001700        BIT     #PGE,@#CS1  ;IS PGE SET
024226  001404                        BEQ     3$          ;BRANCH IF GOOD
024230  013737  001626  001122        MOV     @#RHCS2,@#$BDADR
024236  104037                        ERROR   37          ;PGE DID NOT SET WHEN A WRITE
                                                        ;WAS ATTEMPTED WITH ONE IN PROGRESS
024240                          3$:


;;*****************************************************************
;*      THESE TESTS ARE THROUGH THE MAINTAINABILITY REGISTER
;*      THE SECTOR GAP AND SYNC BYTE ARE ALWAYS READ AS
;*      ZEROS AND 144000 NO MATTER WHAT IS IN THE SIMULATED DISK AREA
;*      TAGED SECGAP: AND WSSYNC:
;*      THE HEADER CONSISTING OF CYLINDER ADDRESS, SECTOR/
;*      TRACK AND THE KEYS ARE READ FROM LOCATION
;*      CYL:, SECTOR:, KEY1:, AND KEY2 AND NOT FROM
;*      HEADER: ON SIMULATED DISK
;*      CRC IS READ FROM SIMULATED DISK LOCATION WCRC:
;*      HEADER GAP IS ALWAYS READ AS ZEROS NO MATTER
;*      WHAT IS ON THE SIMULATED DISK AREA
;*      THE DATA SYNC IS READ  FROM HDWSYN:
;*      ON SIMULATED DISK
;*      ALL DATA IS READ FROM SIMULATED DISK DISK:


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



;;*****************************************************************
;*TEST 57       READ HEADER AND DATA 1
;*      READ CYLINDER 0 FORMAT 16 BITS PER WORD
;*      TRACK 0, SECTOR 0, KEYS 0, 256 WORDS OF 0
;*      ANY ERROR LOGIC INDICATIONS ARE NOT CONCLUSIVE ON FIRST PASS
```

```
                                        ;*      BECAUSE ERROR LOGIC HAS NOT BEEN CHECKED YET (ON FIRST PASS)
                                        ;;*****************************************************************
024240  000004                        TST57:  SCOPE
024242  012706  001000                        MOV     #STACK,SP       ;RESET STACK
024246  012737  000057  004172                MOV     #TTNO,@#TSTNM   ;THIS SAVES TEST NUMBER
                                        ;       SETUP FOR WHAT IS TO BE READ
                                        ;       HEADER CRC IS RESTORED FROM A SUBROUTINE
024254  012746  000000                        MOV     #0,     -(SP)   ;DATA TO BE READ
024260  012705  000400                        MOV     #256.,  R5      ;COUNTER
024264  012700  046330                        MOV     #DISK,  R0      ;START OF SIMULATED DISK DATA
024270  011620                        1$:     MOV     (SP),   (R0)+   ;MOVE IN DATA ON TO SIMULATED DISK
024272  005305                        DEC     R5                      ;COUNT
024274  001375                        BNE     1$                      ;BRANCH IF 256 NOT COMPLETE
024276  005726                        TST     (SP)+                   ;UNDO -(SP)
024300  012705  000021                        MOV     #17.,   R5      ;2 ECC WORDS
                                                                        ;1 DATA GAP
                                                                        ;14 TOLERANCE GAP
024304  005020                        2$:     CLR     (R0)+           ;CLEAR ECC, DATA GAP, AND
024306  005305                        DEC     R5                      ;TOLERANCE GAP
024310  001375                        BNE     2$                      ;BRANCH IF NOT COMPLETE

                                        ;THESE ARE TO SETUP FOR DISKLESS USE ONLY

024312  012737  010000  044412        MOV     #0!FMT22,       @#CYL   ;16 BITS PER WORD
                                                                        ;CYLINDER 0, FORMAT 16 BITS
024320  112737  000000  044415        MOVB    #0,     @#SECOTR+1 ;TRACK 0
024326  112737  000000  044414        MOVB    #0,     @#SECOTR ;SECTOR 0
024334  012737  000000  044416        MOV     #0,     @#KEY1  ;KEY1=0
024342  012737  000000  044420        MOV     #0,     @#KEY2  ;KEY2=0
024350  012737  000400  044472        MOV     #256.,  @#DAWORD ;NO. OF DATA WORDS
024356  005037  044422                CLR     @#X             ;THIS IS A READ COMMAND
024362  004537  041242                JSR     R5,@#CRC        ;GO TO CALCULATE CRC
024366  044412                        CYL
024370  046312                        WCRC

                                        ;THESE ARE REGULAR SETUPS
024372  004737  040064                JSR     PC,@#CLDISK     ;SETUP GENERAL REGISTERS
024376  012777  177374  155216        MOV     #-256.-4.,@RHWC ;256. DATA 4 HEADER WORDS
024404  012777  003126  155212        MOV     #REINTO,@RHBA   ;STARTING ADDRESS OF READ BUFFER
024412  112746  000000                MOVB    #0,     -(SP)   ;IN LOWER BYTE GET SECTOR
024416  112766  000000  000001        MOVB    #0,     1(SP)   ;GET TRACK IN HIGHER BYTE
024424  012677  155204                MOV     (SP)+,  @RHDST  ;TRACK/SECTOR IN RHDST
024430  012777  014000  155202        MOV     #FMT22!ECI,@RHOF ;16 BITS PER WORD
                                                                ;ECC CORRECTION INHIBIT
                                                                ;BECAUSE ECC IS NOT GOING
                                                                ;TO BE CHECKED
024436  005077  155200                CLR     @RHCA           ;CYLINDER 0
```

```
                                ;CHECK FOR DVA,RDY,MOL,DPR,DRY

                        MOV     @#REFOR,@R1     ;READ HEADER AND DATA=72
                        CLR     @#ERFLG$                ;CLEAR ERROR FLAG
                        JSR     PC,     @#COMHD ;READ HEADER AND DATA


                        ;IF THE PROGRAM COMES BACK HERE WITHOUT ERROR PRINTOUTS
                        ;FROM THE "COMHD" ROUTINE THAT MEANS SECTOR GAP,
                        ;FIRST SYNC, HEADER, HEADER CRC, HEADER GAP AND
                        ;SYNC BYTE HAVE GONE BY AND SYNCS WERE CORRECTLY
                        ;DETECTED
                        ;HEADER AND DATA ARE TO BE CHECKED.
                        ;IN CHECKING READ DATA THE WRITE FROM BUFFER
                        ;"WRFROM" IS FILLED WITH EXPECTED DATA AND
                        ;COMPARISONS ARE MADE

                        TST     @#ERFLG$                ;ANY ERRORS ALREADY THERE

                        BNE     TST60   ;BRANCH IF YES

                        JSR     PC,@#CHECKE     ;CHECK DVA,DRY,RDY,DPR
                        MOV     #WRFROM,R0      ;GETTING READY TO FILL EXPECTED DATA
                        MOV     #0!FMT22,(R0)+  ;CYLINDER 0
                        MOVB    #0,     -(SP)   ;IN LOWER BYTE GET SECTOR
                        MOVB    #0,     1(SP)   ;GET TRACK IN HIGHER BYTE
                        MOV     (SP)+,  (R0)+   ;GET TRACK/SECTOR IN BUFFER
                        MOV     #0,     (R0)+   ;KEY1 IN BUFFER
                        MOV     #0,     (R0)+   ;KEY2 IN BUFFER
                        MOV     #256.,  R1      ;DATA WORD COUNTER
                        MOV     #0,     R2      ;DATA
3$:                     MOV     R2,     (R0)+   ;DATA INTO BUFFER
                        DEC     R1              ;COUNT
                        BNE     3$              ;BRANCH IF 256 NOT DONE

                        ;NOW READ DATA BUFFER WILL BE CHECKED

                        JSR     R0,@#COMPAR     ;CHECK
                        WRFROM                  ;GOOD BUFFER
                        REINTO                  ;TEST BUFFER
                        4+256.                  ;NUMBER OF WORDS CHECKED
                        4$                      ;RETURN POINT FOR ERROR HEADER
                        5$                      ;RETURN POINT FOR ERROR DATA

                        TST60                   ;RETURN FOR GOOD COMPARISON

4$:                     ERROR   4               ;READ NEXT ERROR
                        RTS     PC              ;RETURN TO "COMPAR"
5$:                     ERROR   5               ;WORD NOS 1 TO 4 ARE
                                                ;HEADER WORDS
                                                ;5 TO 260 ARE DATA WORDS
                        RTS     PC              ;RETURN TO "COMPAR"
```

```
                                ;;*****************************************************************
                                ;*TEST 60       READ HEADER AND DATA 2
                                ;*      READ CYLINDER 0 FORMAT 16 BITS PER WORD
                                ;*      TRACK 0, SECTOR 1, KEYS 0, 256 WORDS OF 177777
                                ;*      ANY ERROR LOGIC INDICATIONS ARE NOT CONCLUSIVE ON FIRST PASS
                                ;*      BECAUSE ERROR LOGIC HAS NOT BEEN CHECKED YET (ON FIRST PASS)

                                ;;*****************************************************************
024576  000004                  TST60:  SCOPE
024600  012706  001000                  MOV     #STACK,SP       ;RESET STACK

024604  012737  000060  004172          MOV     #TTNO,@#TSTNM   ;THIS SAVES TEST NUMBER


                                ;       SETUP FOR WHAT IS TO BE READ
                                ;       HEADER CRC IS RESTORED FROM A SUBROUTINE

024612  012746  177777                  MOV     #-1,    -(SP)   ;DATA TO BE READ
024616  012705  000400                  MOV     #256.,  R5      ;COUNTER
024622  012700  046330                  MOV     #DISK,  R0      ;START OF SIMULATED DISK DATA
024626  011620                  1$:     MOV     (SP),   (R0)+   ;MOVE IN DATA ON TO SIMULATED DISK
024630  005305                          DEC     R5              ;COUNT
024632  001375                          BNE     1$              ;BRANCH IF 256 NOT COMPLETE
024634  005726                          TST     (SP)+           ;UNDO -(SP)
024636  012705  000021                  MOV     #17.,   R5      ;2 ECC WORDS
                                                                ;1 DATA GAP
                                                                ;14 TOLERANCE GAP
024642  005020                  2$:     CLR     (R0)+           ;CLEAR ECC, DATA GAP, AND
024644  005305                          DEC     R5              ;TOLERANCE GAP
024646  001375                          BNE     2$              ;BRANCH IF NOT COMPLETE


                                ;THESE ARE TO SETUP FOR DISKLESS USE ONLY

024650  012737  010000  044412          MOV     #0!FMT22,       @#CYL   ;16 BITS PER WORD
                                                                        ;CYLINDER 0, FORMAT 16 BITS
024656  112737  000000  044415          MOVB    #0,     @#SECOTR+1 ;TRACK 0
024664  112737  000001  044414          MOVB    #1,     @#SECOTR ;SECTOR 1
024672  012737  000000  044416          MOV     #0,     @#KEY1  ;KEY1=0
024700  012737  000000  044420          MOV     #0,     @#KEY2  ;KEY2=0
024706  012737  000400  044472          MOV     #256.,  @#DAWORD ;NO. OF DATA WORDS
024714  005037  044422                  CLR     @#X             ;THIS IS A READ COMMAND
024720  004537  041242                  JSR     R5,@#CRC        ;GO TO CALCULATE CRC
024724  044412                          CYL
024726  046312                          WCRC


                                ;THESE ARE REGULAR SETUPS
024730  004737  040064                  JSR     PC,@#CLDISK     ;SETUP GENERAL REGISTERS
```

```
6822  024734  012777  177374  154660          MOV     #-256.-4.,@RHWC ;256. DATA 4 HEADER WORDS
6823  024742  012777  003126  154654          MOV     #REINTO,@RHBA   ;STARTING ADDRESS OF READ BUFFER
6824  024750  112746  000001                  MOVB    #1,     -(SP)   ;IN LOWER BYTE GET SECTOR
6825  024754  112766  000000  000001          MOVB    #0,     1(SP)   ;GET TRACK IN HIGHER BYTE
6826  024762  012677  154646                  MOV     (SP)+,  @RHDST  ;TRACK/SECTOR IN RHDST
6827  024766  012777  014000  154644          MOV     #FMT22!ECI,@RHOF ;16 BITS PER WORD
6828                                                                  ;ECC CORRECTION INHIBIT
6829                                                                  ;BECAUSE ECC IS NOT GOING
6830                                                                  ;TO BE CHECKED
6831  024774  005077  154642                  CLR     @RHCA           ;CYLINDER 0
6832
6833  025000  004737  040120                  JSR     PC,     @#CHECKT ;CHECK FOR DVA,RDY,MOL,DPR,DRY
6834
6835  025004  013711  002044                  MOV     @#REFOR,@R1     ;READ HEADER AND DATA=72
6836  025010  005037  001772                  CLR     @#ERFLG$                ;CLEAR ERROR FLAG
6837  025014  004737  044302                  JSR     PC,     @#COMHD ;READ HEADER AND DATA
6838
6839
6840                                          ;IF THE PROGRAM COMES BACK HERE WITHOUT ERROR PRINTOUTS
6841                                          ;FROM THE "COMHD" ROUTINE THAT MEANS SECTOR GAP,
6842                                          ;FIRST SYNC, HEADER, HEADER CRC, HEADER GAP AND
6843                                          ;SYNC BYTE HAVE GONE BY AND SYNCS WERE CORRECTLY
6844                                          ;DETECTED
6845                                          ;HEADER AND DATA ARE TO BE CHECKED.
6846                                          ;IN CHECKING READ DATA THE WRITE FROM BUFFER
6847                                          ;"WRFROM" IS FILLED WITH EXPECTED DATA AND
6848                                          ;COMPARISONS ARE MADE
6849
6850  025020  005737  001772                  TST     @#ERFLG$                ;ANY ERRORS ALREADY THERE
6851
6852  025024  001043                          BNE     TST61   ;BRANCH IF YES
6853
6854  025026  004737  040302                  JSR     PC,@#CHECKE     ;CHECK DVA,DRY,RDY,DPR
6855  025032  012700  002062                  MOV     #WRFROM,R0      ;GETTING READY TO FILL EXPECTED DATA
6856  025036  012720  010000                  MOV     #0!FMT22,(R0)+  ;CYLINDER 0
6857  025042  112746  000001                  MOVB    #1,     -(SP)   ;IN LOWER BYTE GET SECTOR
6858  025046  112766  000000  000001          MOVB    #0,     1(SP)   ;GET TRACK IN HIGHER BYTE
6859  025054  012620                          MOV     (SP)+,  (R0)+   ;GET TRACK/SECTOR IN BUFFER
6860  025056  012720  000000                  MOV     #0,     (R0)+   ;KEY1 IN BUFFER
6861  025062  012720  000000                  MOV     #0,     (R0)+   ;KEY2 IN BUFFER
6862  025066  012701  000400                  MOV     #256.,  R1      ;DATA WORD COUNTER
6863  025072  012702  177777                  MOV     #-1,    R2      ;DATA
6864  025076  010220                  3$:     MOV     R2,     (R0)+   ;DATA INTO BUFFER
6865  025100  005301                          DEC     R1              ;COUNT
6866  025102  001375                          BNE     3$              ;BRANCH IF 256 NOT DONE
6867
6868                                          ;NOW READ DATA BUFFER WILL BE CHECKED
6869
6870  025104  004037  040736                  JSR     R0,@#COMPAR     ;CHECK
6871  025110  002062                          WRFROM                  ;GOOD BUFFER
6872  025112  003126                          REINTO                  ;TEST BUFFER
6873  025114  000404                          4+256.                  ;NUMBER OF WORDS CHECKED
6874  025116  025124                          4$                      ;RETURN POINT FOR ERROR HEADER
6875  025120  025130                          5$                      ;RETURN POINT FOR ERROR DATA
6876
6877  025122  025134                          TST61                   ;RETURN FOR GOOD COMPARISON
```

```
                                4$:     ERROR   4               ;READ NEXT ERROR
                                        RTS     PC              ;RETURN TO "COMPAR"
                                5$:     ERROR   5               ;WORD NOS 1 TO 4 ARE
                                                                ;HEADER WORDS
                                                                ;5 TO 260 ARE DATA WORDS
                                        RTS     PC              ;RETURN TO "COMPAR"


                        ;;**************************************************************
                        ;*TEST 61       READ HEADER AND DATA 3
                        ;*      READ CYLINDER 0 FORMAT 16 BITS PER WORD
                        ;*      TRACK 1, SECTOR 1, KEYS 0, 256 WORDS OF 052525
                        ;*      ANY ERROR LOGIC INDICATIONS ARE NOT CONCLUSIVE ON FIRST PASS
                        ;*      BECAUSE ERROR LOGIC HAS NOT BEEN CHECKED YET (ON FIRST PASS)

                        ;;**************************************************************
025134  000004          TST61:  SCOPE
025136  012706  001000          MOV     #STACK,SP       ;RESET STACK

025142  012737  000061  004172  MOV     #TTNO,@#TSTNM   ;THIS SAVES TEST NUMBER


                        ;       SETUP FOR WHAT IS TO BE READ
                        ;       HEADER CRC IS RESTORED FROM A SUBROUTINE

025150  012746  052525          MOV     #052525,        -(SP)   ;DATA TO BE READ
025154  012705  000400          MOV     #256.,  R5      ;COUNTER
025160  012700  046330          MOV     #DISK,  R0      ;START OF SIMULATED DISK DATA
025164  011620          1$:     MOV     (SP),   (R0)+   ;MOVE IN DATA ON TO SIMULATED DISK
025166  005305                  DEC     R5              ;COUNT
025170  001375                  BNE     1$              ;BRANCH IF 256 NOT COMPLETE
025172  005726                  TST     (SP)+           ;UNDO -(SP)
025174  012705  000021          MOV     #17.,   R5      ;2 ECC WORDS
                                                        ;1 DATA GAP
                                                        ;14 TOLERANCE GAP
025200  005020          2$:     CLR     (R0)+           ;CLEAR ECC, DATA GAP, AND
025202  005305                  DEC     R5              ;TOLERANCE GAP
025204  001375                  BNE     2$              ;BRANCH IF NOT COMPLETE


                        ;THESE ARE TO SETUP FOR DISKLESS USE ONLY

025206  012737  010000  044412  MOV     #0!FMT22,       @#CYL   ;16 BITS PER WORD
                                                                ;CYLINDER 0, FORMAT 16 BITS
025214  112737  000001  044415  MOVB    #1,     @#SECOTR+1 ;TRACK 1
025222  112737  000001  044414  MOVB    #1,     @#SECOTR ;SECTOR 1
025230  012737  000000  044416  MOV     #0,     @#KEY1  ;KEY1=0
025236  012737  000000  044420  MOV     #0,     @#KEY2  ;KEY2=0
```

```
 6934  025244  012737  000400  044472          MOV     #256.,  @#DAWORD ;NO. OF DATA WORDS
 6935  025252  005037  044422                  CLR     @#X              ;THIS IS A READ COMMAND
 6936  025256  004537  041242                  JSR     R5,@#CRC         ;GO TO CALCULATE CRC
 6937  025262  044412                          CYL
 6938  025264  046312                          WCRC
 6939
 6940
 6941                                          ;THESE ARE REGULAR SETUPS
 6942  025266  004737  040064                  JSR     PC,@#CLDISK      ;SETUP GENERAL REGISTERS
 6943  025272  012777  177374  154322          MOV     #-256.-4.,@RHWC  ;256. DATA 4 HEADER WORDS
 6944  025300  012777  003126  154316          MOV     #REINTO,@RHBA    ;STARTING ADDRESS OF READ BUFFER
 6945  025306  112746  000001                  MOVB    #1,     -(SP)    ;IN LOWER BYTE GET SECTOR
 6946  025312  112766  000001  000001          MOVB    #1,     1(SP)    ;GET TRACK IN HIGHER BYTE
 6947  025320  012677  154310                  MOV     (SP)+,  @RHDST   ;TRACK/SECTOR IN RHDST
 6948  025324  012777  014000  154306          MOV     #FMT22!ECI,@RHOF ;16 BITS PER WORD
 6949                                                                   ;ECC CORRECTION INHIBIT
 6950                                                                   ;BECAUSE ECC IS NOT GOING
 6951                                                                   ;TO BE CHECKED
 6952  025332  005077  154304                  CLR     @RHCA            ;CYLINDER 0
 6953
 6954  025336  004737  040120                  JSR     PC,     @#CHECKT ;CHECK FOR DVA,RDY,MOL,DPR,DRY
 6955
 6956  025342  013711  002044                  MOV     @#REFOR,@R1      ;READ HEADER AND DATA=72
 6957  025346  005037  001772                  CLR     @#ERFLG$                 ;CLEAR ERROR FLAG
 6958  025352  004737  044302                  JSR     PC,     @#COMHD  ;READ HEADER AND DATA
 6959
 6960
 6961                                          ;IF THE PROGRAM COMES BACK HERE WITHOUT ERROR PRINTOUTS
 6962                                          ;FROM THE "COMHD" ROUTINE THAT MEANS SECTOR GAP,
 6963                                          ;FIRST SYNC, HEADER, HEADER CRC, HEADER GAP AND
 6964                                          ;SYNC BYTE HAVE GONE BY AND SYNCS WERE CORRECTLY
 6965                                          ;DETECTED
 6966                                          ;HEADER AND DATA ARE TO BE CHECKED.
 6967                                          ;IN CHECKING READ DATA THE WRITE FROM BUFFER
 6968                                          ;"WRFROM" IS FILLED WITH EXPECTED DATA AND
 6969                                          ;COMPARISONS ARE MADE
 6970
 6971  025356  005737  001772                  TST     @#ERFLG$                 ;ANY ERRORS ALREADY THERE
 6972
 6973  025362  001043                          BNE     TST62   ;BRANCH IF YES
 6974
 6975  025364  004737  040302                  JSR     PC,@#CHECKE      ;CHECK DVA,DRY,RDY,DPR
 6976  025370  012700  002062                  MOV     #WRFROM,R0       ;GETTING READY TO FILL EXPECTED DATA
 6977  025374  012720  010000                  MOV     #0!FMT22,(R0)+   ;CYLINDER 0
 6978  025400  112746  000001                  MOVB    #1,     -(SP)    ;IN LOWER BYTE GET SECTOR
 6979  025404  112766  000001  000001          MOVB    #1,     1(SP)    ;GET TRACK IN HIGHER BYTE
 6980  025412  012620                          MOV     (SP)+,  (R0)+    ;GET TRACK/SECTOR IN BUFFER
 6981  025414  012720  000000                  MOV     #0,     (R0)+    ;KEY1 IN BUFFER
 6982  025420  012720  000000                  MOV     #0,     (R0)+    ;KEY2 IN BUFFER
 6983  025424  012701  000400                  MOV     #256.,  R1       ;DATA WORD COUNTER
 6984  025430  012702  052525                  MOV     #052525,R2       ;DATA
 6985  025434  010220          3$:             MOV     R2,     (R0)+    ;DATA INTO BUFFER
 6986  025436  005301                          DEC     R1               ;COUNT
 6987  025440  001375                          BNE     3$               ;BRANCH IF 256 NOT DONE
 6988
 6989                                          ;NOW READ DATA BUFFER WILL BE CHECKED
```

```
                                                ;CHECK
025442  004037  040736                  JSR     R0,@#COMPAR     ;GOOD BUFFER
025446  002062                          WRFROM                  ;TEST BUFFER
025450  003126                          REINTO                  ;NUMBER OF WORDS CHECKED
025452  000404                          4+256.                  ;RETURN POINT FOR ERROR HEADER
025454  025462                          4$                      ;RETURN POINT FOR ERROR DATA
025456  025466                          5$
                                                ;RETURN FOR GOOD COMPARISON
025460  025472                          TST62
                                4$:     ERROR   4               ;READ NEXT ERROR
025462  104004                          RTS     PC              ;RETURN TO "COMPAR"
025464  000207                  5$:     ERROR   5               ;WORD NOS 1 TO 4 ARE
025466  104005                                                  ;HEADER WORDS
                                                                ;5 TO 260 ARE DATA WORDS
                                        RTS     PC              ;RETURN TO "COMPAR"
025470  000207
```

```
                        ;;*****************************************************************
                        ;*TEST 62        WRITE DATA

                        ;*      WRITE CYLINDER 0, FORMAT 16 BITS PER WORD
                        ;*      TRACK 0, SECTOR 0, KEYS 0, NUMBER OF WORDS 256 OF 377
                        ;*      ANY ERROR LOGIC INDICATION IS NOT CONCLUSIVE ON FIRST PASS
                        ;*      BECAUSE ERROR LOGIC HAS NOT BEEN CHECKED

                        ;;*****************************************************************
025472  000004         TST62:  SCOPE

025474  012706  001000                  MOV     #STACK,SP       ;RESET STACK

025500  012737  000062  004172          MOV     #TTNO,@#TSTNM   ;THIS SAVES TEST NUMBER

025506  004037  040002                  JSR     R0,@#CLAREA     ;CLEAR SIMULATED DISK
025512  046330                          .WORD   DISK            ;FROM
025514  047354                          .WORD   TOLGAP+16       ;TO
025516  000000                          .WORD   0               ;DATA
                        ;THESE ARE SETUP FOR DISKLESS USE ONLY
025520  012737  010000  044412          MOV     #0!FMT22,@#CYL;CYLINDER 0
                                                                ;16 BITS PER WORD
025526  112737  000000  044415          MOVB    #0,@#SECOTR+1   ;TRACK 0
025534  112737  000000  044414          MOVB    #0,@#SECOTR     ;SECTOR 0
025542  005037  044416                  CLR     @#KEY1          ;KEY1 0
025546  005037  044420                  CLR     @#KEY2          ;KEY2 0
025552  012737  000400  044460          MOV     #256.,@#NOWORD  ;NO OF DATA WORDS
025560  012737  000001  044422          MOV     #1,@#X          ;WRITE DATA
025566  004537  041242                  JSR     R5,@#CRC        ;GO TO CALCULATE CRC
025572  044412                          CYL
025574  046312                          WCRC
```

```
                                        ;THESE ARE REGULAR SETUPS


025576  004037  040002                    JSR     R0,@#CLAREA     ;FILL WRITE BUFFER WITH 377
025602  002062                            WRFROM                  ;FROM
025604  003062                            WRFROM+<256.*2>         ;TO
025606  000377                            377                     ;DATA
025610  004737  040064                    JSR     PC,@#CLDISK     ;SETUP GENERAL REGISTERS
025614  012777  177400  154000            MOV     #-256.,@RHWC    ;256. DATA WORDS
025622  012777  002062  153774            MOV     #WRFROM,@RHBA   ;STARTING ADDRESS OF WRITE BUFFER
025630  012746  000000                    MOV     #0,-(SP)        ;SECTOR 0
025634  112766  000000  000001            MOVB    #0,1(SP)        ;TRACK 0
025642  012677  153766                    MOV     (SP)+,@RHDST    ;SECTOR 0 TRACK 0
025646  012777  010000  153764            MOV     #FMT22,@RHOF    ;16 BITS PER WORD FORMAT
025654  012777  000000  153760            MOV     #0,@RHCA        ;CYLINDER 0
025662  004737  040120                    JSR     PC,@#CHECKT     ;CHECK FOR DVA,RDY,DPR,DRY
025666  013711  002036                    MOV     @#WRIDAT,@R1    ;WRITE DATA=60
025672  005037  001772                    CLR     @#ERFLG$        ;CLEAR ERROR FLAG
025676  004737  044302                    JSR     PC,@#COMHD      ;WRITE DATA
                                        ;IF THE PROGRAM COMES BACK HERE WITHOUT ERROR PRINTOUTS
                                        ;FROM THE "COMHD" ROUTINE IT MEANS SECTOR GAP, SYNC BYTE
                                        ;HEADER, HEADER CRC, HEADER GAP AND SYNC BYTE HAVE GONE BY
                                        ;AND SYNCS WERE CORRECTLY DETECTED
                                        ;DATA IS TO BE CHECKED
025702  004737  037470                    JSR     PC,@#PUTREG     ;SAVE REGISTERS
025706  005737  001772                    TST     @#ERFLG$                ;HAS ANY ERRORS OCCURED?

025712  001041                            BNE     TST63   ;;BRANCH IF YES

025714  012700  000377                    MOV     #377,R0         ;GOOD DATA
025720  012701  046330                    MOV     #DISK,R1        ;DATA WRITTEN INTO "DISK"
025724  012702  000400                    MOV     #256.,R2                ;COUNTER
025730  012737  000401  044532  1$:       MOV     #256.+1,@#ERWORD        ;FOR ERROR WORD
025736  020021                            CMP     R0,(R1)+        ;COMPARE GOOD DATA WITH DATA ON DISK
025740  001424                            BEQ     3$              ;BRANCH IF GOOD
025742  010037  001124                    MOV     R0,@#$GDDAT     ;GOOD DATA
025746  014137  001126                    MOV     -(R1),@#$BDDAT  ;BAD DATA
025752  160237  044532                    SUB     R2,@#ERWORD     ;ERROR WORD NO
025756  005737  001772                    TST     @#ERFLG$                ;ANY ERRORS ALREADY THERE?
025762  001002                            BNE     2$              ;BRANCH IF YES
025764  104004                            ERROR   4               ;ERROR ON WRITE DATA COMMAND
025766  000401                            BR      64$             ;BRANCH TO AVOID PRINTING NEXT ERROR
025770  104005                  2$:       ERROR   5               ;WORD NO GIVES WORD IN ERROR
025772  005721                  64$:      TST     (R1)+           ;UNDO -(R1) FOR BAD DATA
025774  013746  001140                    MOV     @#SWR,-(SP)     ;GET SWITCH SETTING
026000  042716  177177                    BIC     #177177,(SP)    ;KEEP ONLY SWITCH 7 AND 8
026004  022726  000200                    CMP     #SW07,(SP)+     ;IS 7 SET AND 8 RESET

026010  001402                            BEQ     TST63   ;BRANCH OUT IF YES

026012  005302                  3$:       DEC     R2              ;IF NOT COUNT 256 WORDS
026014  001345                            BNE     1$              ;BRANCH IF 256. NOT DONE
```

```
                                ;;*****************************************************************
                                ;*TEST 63       READ DATA

                                ;*      READ CYLINDER0, FORMAT 16 BITS PER WORD
                                ;*      TRACK0, SECTOR 1, KEYS 0, 10 WORDS OF 177400
                                ;*      ANY ERROR LOGIC INDICATION IS NOT CONCLUSIVE
                                ;*      BECAUSE ERROR LOGIC HAS NOT BEEN CHECKED YET (ON FIRST PASS)

                                ;;*****************************************************************
026016  000004                  TST63:  SCOPE
026020  012706  001000                  MOV     #STACK,SP       ;RESET STACK

026024  012737  000063  004172          MOV     #TTNO,@#TSTNM   ;THIS SAVES TEST NUMBER

026032  004037  040002                  JSR     R0,@#CLAREA     ;CLEAR SIMULATED DISK
026036  046330                          .WORD   DISK            ;FROM
026040  047326                          .WORD   DISK+776        ;TO
026042  177400                          .WORD   177400          ;DATA

026044  004037  040002                  JSR     R0,@#CLAREA     ;CLEAR READ INTO BUFFER
026050  003126                          .WORD   REINTO          ;FROM
026052  004124                          .WORD   REINTO+776      ;TO
026054  000000                          .WORD   0               ;DATA

                                ;THESE ARE TO SETUP FOR DISKLESS USE ONLY

026056  012737  010000  044412          MOV     #FMT22,@#CYL    ;CYLINDER 0 16 BITS PER WORD FORMAT
026064  105037  044415                  CLRB    @#SECOTR+1      ;TRACK 0
026070  112737  000001  044414          MOVB    #1,@#SECOTR     ;SECTOR 1
026076  005037  044416                  CLR     @#KEY1          ;KEY1=0
026102  005037  044420                  CLR     @#KEY2          ;KEY2=0
026106  012737  000012  044472          MOV     #10.,@#DAWORD   ;NO. OF DATA WORDS
026114  005037  044422                  CLR     @#X             ;THIS IS A READ COMMAND
026120  004537  041242                  JSR     R5,@#CRC        ;GO TO CALCULATE CRC
026124  044412                          CYL
026126  046312                          WCRC

                                ;THESE ARE REGULAR SETUPS

026130  004737  040064                  JSR     PC,@#CLDISK     ;SETUP GENERAL REGISTERS
026134  013711  002042                  MOV     @#READAT,@R1    ;READ DATA INTO RHCS1=70
026140  012777  177766  153454          MOV     #-10.,@RHWC     ;10 DATA WORDS
026146  012777  003126  153450          MOV     #REINTO,@RHBA   ;STARTING ADDRESS OF READ BUFFER
026154  112746  000001                  MOVB    #1,-(SP)        ;IN LOWER BYTE GET SECTOR 1
026160  112766  000000  000001          MOVB    #0,1(SP)        ;GET TRACK0 IN UPPER BYTE
026166  012677  153442                  MOV     (SP)+,@RHDST    ;TRACK/SECTOR IN RHDST
026172  012777  014000  153440          MOV     #FMT22!ECI,@RHOF ;16 BITS PER WORD
                                                                ;ECC CORRECTION INHIBIT BECAUSE
                                                                ;ECC IS NOT CHECKED HERE
026200  005077  153436                  CLR     @RHCA           ;CYLINDER 0
026204  004737  040120                  JSR     PC,@#CHECKT     ;CHECK FOR DVA,RDY,MOL,DPR,DRY
026210  005037  001772                  CLR     @#ERFLG$        ;CLEAR ERROR FLAG
026214  004737  044302                  JSR     PC,@#COMHD      ;READ DATA
```

```
                                        ;IF THE PROGRAM COMES BACK HERE WITHOUT ERROR PRINTOUT
                                        ;FROM "COMHD" ROUTINE IN MEANS DATA IS TO BE CHECKED

                                        ;NOW THE DATA READ INTO "REINTO" BUFFER WILL
                                        ;BE CHECKED, ONLY 10 WORDS SHOULD BE CHANGED
                                        ;ALL OTHER WORDS SHOULD REMAIN UNCHANGED
                                        ;THE "WRFROM" BUFFER IS FILLED WITH EXPECTED DATA AND CHECKED

026220  005737  001772                  TST     @#ERFLG$                ;HAS ANY ERRORS OCCURED?

026224  001053                          BNE     TST64   ;;BRANCH IF YES

026226  004037  040002                  JSR     R0,@#CLAREA     ;CLEAR BUFFER
026232  002062                          WRFROM                  ;FROM
026234  003060                          WRFROM+776              ;TO
026236  000000                          0                       ;DATA

026240  004037  040002                  JSR     R0,@#CLAREA     ;FILL EXPECTED DATA
026244  002062                          WRFROM                  ;FROM
026246  002104                          WRFROM+22               ;TO
026250  177400                          177400                  ;DATA

                                        ;NOW READ DATA BUFFER IS CHECKED

026252  012700  002062                  MOV     #WRFROM,R0      ;GOOD DATA
026256  012701  003126                  MOV     #REINTO,R1      ;DATA READ
026262  012702  000400                  MOV     #256.,R2        ;COUNTER
026266  012737  000401  044532  1$:     MOV     #257.,@#ERWORD  ;FOR ERROR WORD NO
026274  022021                          CMP     (R0)+,(R1)+     ;COMPARE GOOD WITH READ BUFFER
026276  001424                          BEQ     2$              ;BRANCH IF GOOD
026300  014037  001124                  MOV     -(R0),@#$GDDAT  ;GOOD DATA
026304  014137  001126                  MOV     -(R1),@#$BDDAT  ;BAD DATA
026310  160237  044532                  SUB     R2,@#ERWORD     ;ERROR WORD NO
026314  005737  001772                  TST     @#ERFLG$                ;ANY ERRORS ALREADY THERE
026320  001002                          BNE     3$              ;IF YES BRANCH DO NOT TYPE HEADER
026322  104004                          ERROR   4               ;ERROR ON READ DATA
026324  000401                          BR      4$              ;BRANCH TO AVOID PRINTING NEXT ERROR
026326  104005                  3$:     ERROR   5               ;WORD NO 1-10 ARE DATA
                                                                ;WORDS
                                                                ;WORD NOS 11-256 HAVE NOT BEEN
                                                                ;READ AND BUFFER SHOULD BE
                                                                ;ZERO IF OTHER THAN ZERO
                                                                ;WRONG NUMBER OF WORDS HAVE
                                                                ;BEEN READ IN THE DISK NOW
                                                                ;CONTAINS 177400 ALL 256
                                                                ;WORDS BUT ONLY 10 WORDS
                                                                ;SHOULD BE READ IN

026330  022021                  4$:     CMP     (R0)+,(R1)+     ;UNDO -(R0) AND -(R1) FOR ERROR
026332  013746  001140                  MOV     @#SWR,-(SP)     ;GET SWITCH SETTING
026336  042716  177177                  BIC     #177177,(SP)    ;KEEP ONLY SWITCH 7 AND 8
026342  022726  000200                  CMP     #SW07,(SP)+     ;IS 7 SET AND 8 RESET

026346  001402                          BEQ     TST64   ;BRANCH OUT IF YES

026350  005302                  2$:     DEC     R2              ;COUNT
```

```
7214  026352  001345                    BNE     1$              ;BRANCH IF NOT COMPLETE
7215
7216
7217
7218
7219                                  ;
7220
7221                                  ;;*****************************************************************
7222                                  ;*TEST 64       WRITE CHECK HEADER AND DATA
7223
7224                                  ;*      WRITE CHECK CYLINDER 0, FORMAT 16 BITS PER WORD
7225                                  ;*      TRACK 1, SECTOR 1, KEYS 0, 36 WORDS AS SHOWN BELOW
7226                                  ;*      ANY DEVICE LOGIC ERROR INDICATION IS NOT CONCLUSIVE ON FIRST PASS
7227                                  ;*      BECAUSE ERROR LOGIC HAS NOT BEEN CHECKED YET (ON FIRST PASS)
7228                                  ;*      ONLY RH WRITE CHECK ERROR (RHCS2 BIT 14) IS TESTED HERE
7229                                  ;;*****************************************************************
7230  026354  000004                  TST64:  SCOPE
7231                                  ;       DATA TABLE
7232                                  ;       20 WORDS OF 070707
7233                                  ;       THEN 16 WORDS WITH ZERO FLOATING FROM RIGHT
7234                                  ;       TO LEFT (EG. 177776,177775,177773 ETC)
7235
7236
7237  026356  012706  001000                  MOV     #STACK,SP       ;RESET STACK
7238
7239
7240  026362  012737  000064  004172          MOV     #TTNO,@#TSTNM   ;THIS SAVES TEST NUMBER
7241
7242
7243                                          ;SET UP "REINTO" FOR WHAT IS TO BE READ
7244  026370  012701  003126                  MOV     #REINTO,R1      ;STARTING ADDRESS
7245  026374  012721  010000                  MOV     #FMT22,(R1)+    ;CYLINDER 0 FORMAT 16 BIT WORDS
7246  026400  012721  000401                  MOV     #401,(R1)+      ;TRACK=1, SECTOR=1
7247  026404  005021                          CLR     (R1)+           ;KEY1=0
7248  026406  005021                          CLR     (R1)+           ;KEY2=0
7249  026410  004037  040002                  JSR     R0,@#CLAREA     ;FILL "REINTO" BUFFER
7250  026414  003136                          .WORD   REINTO+<4*2>    ;FROM
7251  026416  003204                          .WORD   REINTO+<23.*2>  ;TO
7252  026420  070707                          .WORD   070707          ;DATA
7253
7254  026422  012700  177776                  MOV     #177776,R0      ;GETTING READY TO FLOAT 0
7255  026426  012701  003206                  MOV     #REINTO+<24.*2>,R1;STARTING ADDRESS WHERE 177776 GOES
7256  026432  010021                  1$:     MOV     R0,(R1)+        ;MOVE IN FLOATING 0
7257  026434  000261                          SEC                     ;SET CARRY
7258  026436  006100                          ROL     R0              ;GET 0 ONE BIT LEFT
7259  026440  103774                          BCS     1$              ;BRANCH IF 16 NOT DONE
7260
7261  026442  004037  040002                  JSR     R0,@#CLAREA     ;FILL THE REST OF BUFFER WITH 0
7262  026446  003246                          .WORD   REINTO+<40.*2>  ;FROM
7263  026450  004124                          .WORD   REINTO+776      ;TO
7264  026452  000000                          .WORD   0               ;DATA
7265
7266                                          ;SET UP SIMULATED DISK WITH WHAT IS TO BE READ
7267
7268  026454  004037  040002                  JSR     R0,@#CLAREA     ;FILL "DISK" BUFFER
7269  026460  046330                          .WORD   DISK            ;FROM
```

```
7270  026462  046376                          .WORD   DISK+(19.*2)    ;TO
7271  026464  070707                          .WORD   070707          ;DATA
7272
7273  026466  012700  177776                  MOV     #177776,R0      ;GETTING READY TO FLOAT ZEROS
7274  026472  012701  046400                  MOV     #DISK+(20.*2),R1        ;STARTING ADDRESS WHERE 177776 GOES
7275  026476  010021                  2$:     MOV     R0,(R1)+        ;MOVE IN FLOATING 0
7276  026500  000261                          SEC                     ;SET CARRY
7277  026502  006100                          ROL     R0              ;GET 0 ONE BIT LEFT
7278  026504  103774                          BCS     2$              ;BRANCH IF 16 NOT DONE
7279
7280  026506  004037  040002                  JSR     R0,@#CLAREA     ;FILL THE REST OF BUFFER WITH 177777
7281  026512  046440                          .WORD   DISK+(36.*2)    ;FROM
7282  026514  047326                          .WORD   DISK+776        ;TO
7283  026516  177777                          .WORD   177777          ;DATA
7284
7285  026520  004737  040574                  JSR     PC,@#WRCHHD     ;WRITE CHECK HEADER AND DATA
7286                                                                   ;CYLINDER 0, TRACK 1, SECTOR 1
7287
7288                                           ;IF THE PROGRAM COMES BACK HERE THEN WRITE CHECK
7289                                           ;HAS BEEN COMPLETED NOW WRITE CHECK ERROR BIT IS TO BE TESTED
7290  026524  013746  001760                  MOV     @#UNIT,-(SP)    ;GET UNIT NUMBER
7291  026530  052716  000100                  BIS     #IR,(SP)        ;ONLY BIT 6 SHOULD BE SET
7292  026534  004737  037470                  JSR     PC,@#PUTREG     ;SAVE REGISTERS
7293  026540  022637  001676                  CMP     (SP)+,@#CS2     ;COMPARE RHCS2
7294  026544  001406                          BEQ     4$              ;BRANCH IF GOOD
7295  026546  032712  040000                  BIT     #WCE,@R2        ;WRITE CHECK ERROR HIGH?
7296  026552  001402                          BEQ     3$              ;BRANCH IF ERROR NOT DUE TO "WCE"
7297  026554  104017                          ERROR   17              ;RHDB CONTAINS FAILING WORD
7298  026556  000401                          BR      4$              ;RHBA CONTAINS ADDRESS+2
7299                                                                   ;OF THE WORD IN MEMORY FROM
7300                                                                   ;THE DISK THAT DID NOT COMPARE
7301                                                                   ;TRE AND SC WILL BE SET DUE TO
7302                                                                   ;WCE
7303  026560  104017                  3$:     ERROR   17              ;WCE CORRECTLY WAS NOT SET BUT SOME
7304                                                                   ;BITS OTHER THAN IR
7305                                                                   ;AND UNIT NO. WAS SET
7306
7307                                           ;NOW CHECK MEMORY TO SEE IF NOTHING GOT DESTROYED
7308                                           ;FILL "WRFROM" WITH WHAT SHOULD BE IN "REINTO" THEN CHECK
7309  026562  012700  002062          4$:     MOV     #WRFROM,R0      ;STARTING ADDRESS
7310  026566  012720  010000                  MOV     #FMT22,(R0)+    ;CYLINDER
7311  026572  012720  000401                  MOV     #401,(R0)+      ;TRACK=1, SECTOR=1
7312  026576  005020                          CLR     (R0)+           ;KEY1=0
7313  026600  005020                          CLR     (R0)+           ;KEY2=0
7314
7315  026602  004037  040002                  JSR     R0,@#CLAREA     ;FILL "WRFROM" BUFFER
7316  026606  002072                          .WORD   WRFROM+(4*2)    ;FROM
7317  026610  002140                          .WORD   WRFROM+(23.*2)  ;TO
7318  026612  070707                          .WORD   070707          ;DATA
7319
7320  026614  012700  177776                  MOV     #177776,R0      ;GETTING READY TO FLOAT 0
7321  026620  012701  002142                  MOV     #WRFROM+(24.*2),R1 ;STARTING ADDRESS WHERE 177776 GOES
7322  026624  010021                  5$:     MOV     R0,(R1)+        ;MOVE IN FLOATING 0
7323  026626  000261                          SEC                     ;SET CARRY
7324  026630  006100                          ROL     R0              ;GET 0 ONE BIT LEFT
7325  026632  103774                          BCS     5$              ;BRANCH IF 16 NOT DONE
```

```
7326
7327  026634  004037  040002          JSR     R0,@#CLAREA     ;FILL THE REST OF BUFFER WITH 0
7328  026640  002202                  .WORD   WRFROM+<40.*2>  ;FROM
7329  026642  003060                  .WORD   WRFROM+776      ;TO
7330  026644  000000                  .WORD   0               ;DATA
7331
7332                                  ;NOW THE READ BUFFER WILL BE CHECKED
7333  026646  005037  001772          CLR     @#ERFLG$                ;CLEAR ERROR FLAG
7334
7335  026652  004037  040736          JSR     R0,@#COMPAR     ;CHECK
7336  026656  002062                  WRFROM                  ;GOOD BUFFER
7337  026660  003126                  REINTO                  ;TEST BUFFER
7338  026662  000400                  256.                    ;NUMBER OF WORDS CHECKED
7339  026664  026672                  6$                      ;RETURN POINT FOR ERROR HEADER
7340  026666  026676                  7$                      ;RETURN POINT FOR ERROR DATA
7341
7342  026670  026704                  TST65                   ;RETURN FOR GOOD COMPARISON
7343
7344  026672  104004          6$:     ERROR   4               ;READ NEXT ERROR 5
7345  026674  000207                  RTS     PC              ;RETURN TO COMPARISON SUBROUTINE
7346  026676  104005          7$:     ERROR   5               ;DATA IN REINTO BUFFER GOT
7347                                                          ;CHANGED AFTER A WRITE
7348                                                          ;CHECK HEADER AND DATA COMMAND
7349                                                          ;WORD NO CONTAINS THE WORD
7350                                                          ;NUMBER THAT GOT CHANGED
7351  026700  000207                  RTS     PC              ;RETURN TO COMPARISON SUBROUTINE
7352
7353  026702  000240          10$:    NOP                     ;ONLY A BRANCH POINT
7354
7355                          ;;**************************************************************
7356                          ;*TEST 65       WRITE CHECK DATA
7357
7358                          ;*      WRITE CHECK DATA CYLINDER 0 FORMAT 16 BITS PER WORD
7359                          ;*      TRACK 1, SECTOR 1, KEYS 0, 32 WORDS OF DATA
7360                          ;*      ANY DEVICE LOGIC ERROR INDICATIONS ARE NOT CONCLUSIVE ON FIRST PASS
7361                          ;*      BECAUSE ERROR LOGIC HAS NOT YET (ON FIRST PASS) BEEN CHECKED
7362                          ;*      ONLY RH WRITE CHECK ERROR IS TESTED
7363
7364                          ;;**************************************************************
7365  026704  000004          TST65:  SCOPE
7366
7367                                  ;DATA TABLE
7368                                  ;TOTAL OF 32 WORDS CONSISTING OF
7369                                  ;16 WORDS OF FLOATING ONES (EG. 1, 2, 4, 10)
7370                                  ;16 WORDS OF FLOATING ZEROS (EG. 177776, 177775)
7371
7372
7373  026706  012706  001000          MOV     #STACK,SP       ;RESET STACK
7374
7375
7376  026712  012737  000065  004172  MOV     #TTNO,@#TSTNM   ;THIS SAVES TEST NUMBER
7377
7378                                  ;SET UP "REINTO" FOR WHAT IS TO BE READ
7379
7380  026720  012700  000001          MOV     #1,R0           ;GETTING READY TO FLOAT 1
7381  026724  012701  003126          MOV     #REINTO,R1      ;STARTING ADDRESS WHERE 1 GOES
```

```
7382  026730  010021                1$:   MOV     R0,(R1)+          ;MOVE FLOATING 1
7383  026732  006100                      ROL     R0                ;GET 1 ONE BIT LEFT
7384  026734  103375                      BCC     1$                ;BRANCH IF 16 NOT DONE
7385  026736  012700  177776              MOV     #177776,R0        ;GETTING READY TO FLOAT 0
7386  026742  012701  003166              MOV     #REINTO+<16.*2>,R1 ;STARTING ADDRESS WHERE 177776 GOES
7387  026746  010021                2$:   MOV     R0,(R1)+          ;MOVE IN FLOATING 0
7388  026750  000261                      SEC                       ;SET CARRY
7389  026752  006100                      ROL     R0                ;GET 0 ONE BIT LEFT
7390  026754  103774                      BCS     2$                ;BRANCH IF 16 NOT DONE
7391
7392  026756  004037  040002              JSR     R0,@#CLAREA       ;FILL REST OF BUFFER WITH 1
7393  026762  003226                      .WORD   REINTO+<32.*2>    ;FROM
7394  026764  004124                      .WORD   REINTO+776        ;TO
7395  026766  000001                      .WORD   1                 ;WITH DATA
7396
7397                                      ;SET UP SIMULATED DISK WITH WHAT IS TO BE READ
7398
7399  026770  012700  000001              MOV     #1,R0             ;GETTING READY TO FLOAT 1
7400  026774  012701  046330              MOV     #DISK,R1          ;STARTING ADDRESS WHERE 1 GOES
7401  027000  010021                3$:   MOV     R0,(R1)+          ;MOVE FLOATING 1
7402  027002  006100                      ROL     R0                ;GET 1 ONE BIT LEFT
7403  027004  103375                      BCC     3$                ;BRANCH IF 16 NOT DONE
7404
7405  027006  012700  177776              MOV     #177776,R0        ;GETTING READY TO FLOAT 0
7406  027012  012701  046370              MOV     #DISK+<16.*2>,R1  ;STARTING ADDRESS WHERE 177776 GOES
7407  027016  010021                4$:   MOV     R0,(R1)+          ;MOVE FLOATING 0
7408  027020  000261                      SEC                       ;SET CARRY
7409  027022  006100                      ROL     R0                ;GET 0 ONE BIT LEFT
7410  027024  103774                      BCS     4$                ;BRANCH IF 16 NOT DONE
7411
7412  027026  004037  040002              JSR     R0,@#CLAREA       ;FILL REST OF BUFFER WITH 0
7413  027032  046430                      .WORD   DISK+<32.*2>      ;FROM
7414  027034  047326                      .WORD   DISK+776          ;TO
7415  027036  000000                      .WORD   0                 ;WITH DATA
7416
7417  027040  004737  041100              JSR     PC,@#WRCHDA       ;WRITE CHECK DATA
7418                                                                 ;CYLINDER 0, TRACK 1, SECTOR 1
7419                                                                 ;KEYS 0, 32 WORDS.
7420
7421                                      ;IF THE PROGRAM COMES BACK HERE THEN WRITE CHECK
7422                                      ;HAS BEEN COMPLETED NOW WRITE CHECK ERROR BIT IS TESTED
7423
7424  027044  013746  001760              MOV     @#UNIT,-(SP)      ;GET UNIT NUMBER
7425  027050  052716  000100              BIS     #IR,(SP)          ;ONLY BIT 6 SHOULD BE SET
7426  027054  004737  037470              JSR     PC,@#PUTREG       ;SAVE REGISTERS
7427  027060  022637  001676              CMP     (SP)+,@#CS2       ;COMPARE RHCS2
7428  027064  001407                      BEQ     6$                ;BRANCH IF GOOD
7429  027066  032737  040000  001676      BIT     #WCE,@#CS2        ;WRITE CHECK ERROR HIGH?
7430  027074  001402                      BEQ     5$                ;BRANCH IF ERROR NOT DUE TO "WCE"
7431  027076  104017                      ERROR   17                ;RHDB CONTAINS FAILING WORD
7432  027100  000401                      BR      6$                ;RHBA CONTAINS ADDRESS+2
7433                                                                 ;OF THE WORD IN MEMORY FROM
7434                                                                 ;THE DISK THAT DID NOT COMPARE
7435                                                                 ;TRE AND SC WILL BE SET DUE TO WCE
7436  027102  104017                5$:   ERROR   17                ;WCE WAS CORRECTLY NOT SET
7437                                                                 ;BUT SOME BITS OTHER THAN
```

```
                                                        ;IR AND UNIT NO. WERE SET

                                                ;NOW CHECK MEMORY TO SEE IF ANYTHING GOT DESTROYED
                                                ;FILL "WRFROM" WITH WHAT SHOULD BE IN REINTO THEN CHECK

027104  005037  001772        6$:     CLR     @#ERFLG$        ;CLEAR ERROR FLAG
027110  012700  000001                MOV     #1,R0           ;GETTING READY TO FLOAT 1
027114  012701  002062                MOV     #WRFROM,R1      ;START ADDRESS WHERE 1 GOES
027120  010021                7$:     MOV     R0,(R1)+        ;MOVE FLOATING 1
027122  006100                        ROL     R0              ;GET 1 ONE BIT LEFT
027124  103375                        BCC     7$              ;BRANCH IF 16 NOT DONE

027126  012700  177776                MOV     #177776,R0      ;GETTING READY TO FLOAT 0
027132  012701  002122                MOV     #WRFROM+<16.*2>,R1 ;STARTING ADDRESS WHERE 177776 GOES
027136  010021                10$:    MOV     R0,(R1)+        ;MOVE IN FLOATING 0
027140  000261                        SEC                     ;SET CARRY
027142  006100                        ROL     R0              ;GET 0 ONE BIT LEFT
027144  103774                        BCS     10$             ;BRANCH IF CARRY SET

027146  004037  040002                JSR     R0,@#CLAREA     ;FILL REST OF BUFFER WITH 1
027152  002162                        .WORD   WRFROM+<32.*2>  ;FROM
027154  003060                        .WORD   WRFROM+776      ;TO
027156  000001                        .WORD   1               ;WITH DATA

                                                ;NOW THE READ BUFFER WILL BE CHECKED

027160  004037  040736                JSR     R0,@#COMPAR     ;CHECK
027164  002062                        WRFROM                  ;GOOD BUFFER
027166  003126                        REINTO                  ;TEST BUFFER
027170  000400                        256.                    ;NUMBER OF WORDS CHECKED
027172  027200                        11$                     ;RETURN POINT FOR ERROR HEADER
027174  027204                        12$                     ;RETURN POINT FOR ERROR DATA

027176  027212                        TST66                   ;RETURN FOR GOOD COMPARISON

027200  104004                11$:    ERROR   4               ;READ NEXT ERROR 5
027202  000207                        RTS     PC              ;RETURN TO COMPARISON SUBROUTINE
027204  104005                12$:    ERROR   5               ;DATA IN REINTO BUFFER GOT
                                                ;CHANGED AFTER A WRITE
                                                ;CHECK DATA COMMAND
                                                ;WORD NO CONTAINS THE WORD
                                                ;NUMBER THAT GOT CHANGED
027206  000207                        RTS     PC              ;RETURN TO COMPARISON SUBROUTINE

027210  000240                13$:    NOP                     ;ONLY A BRANCH POINT


                        ;;**************************************************************
                        ;*TEST 66        ATTENTION WITH ERROR TEST

                        ;*      THIS TESTS THE SETTING OF ATA BIT BOTH IN THE RHAS
                        ;*      AND THE RHDS1 REGISTERS WITH THE SETTING OF EACH
```

```
                                        ;*      ERROR BIT ON THE THREE ERROR REGISTERS.
                                        ;*      IN EACH OF THE ABOVE CASES ERR IN RHDS1 SHOULD
                                        ;*      ALSO SET
                                        ;*      "GO" SHOULD CLEAR ERR, ATA IN RHDS1 AND RHAS BUT NOT ERROR REG.
                                        ;*      PUTTING "1" IN RHAS DRIVE POSITION CLEARS DRIVE BIT IN ATA IN RHDS1
                                        ;*      UPPER BYTE OF RHAS IS INVALIBD

                                        ;;*****************************************************************
027212  000004                    TST66:  SCOPE

027214  012706  001000                    MOV     #STACK,SP       ;RESET STACK

027220  012737  000066  004172            MOV     #TTNO,@#TSTNM   ;THIS SAVES TEST NUMBER

027226  004737  040064                    JSR     PC,@#CLDISK     ;CLEAR DISK REGISTERS
027232  012700  003126                    MOV     #REINTO,R0      ;BUFFER STARTING FOR 3 ERROR
                                                                  ;REGISTERS
027236  013720  001632                    MOV     @#RHER1,(R0)+   ;RHER1 STORED IN REINTO
027242  012720  000000                    MOV     #0,(R0)+        ;BITS NOT TO BE CHECKED IN RHER1
027246  013720  001636                    MOV     @#RHER2,(R0)+   ;RHER2 STORED IN REINTO+4
027252  012720  040000                    MOV     #BIT14,(R0)+    ;BIT14 NOT TO BE CHECKED IN RHER2
027256  013720  001644                    MOV     @#RHER3,(R0)+   ;RHER3 STORED IN REINTO+10
027262  012720  037600                    MOV     #BIT7!BIT8!BIT9!BIT10!BIT11!BIT12!BIT13,(R0)+
                                                                  ;THE ABOVE BITS ARE NOT TO BE CHECKED
                                                                  ;IN TESTING RHER3
027266  013704  001646                    MOV     @#RHAS,R4       ;R4 HAS RHAS
027272  013705  002002                    MOV     @#ATTENT,R5     ;R5 HAS ATA BIT IN RHAS
027276  012737  027324  001110            MOV     #2$,@#$LPERR    ;THAT SHOULD SET WITH ERROR
                                                                  ;RETURN POINT TO ERROR
027304  012737  000003  001200            MOV     #3,@#$TMP1      ;COUNTER
027312  012700  003126                    MOV     #REINTO,R0      ;POINTER
027316  012002                    1$:     MOV     (R0)+,R2        ;R2 HAS ADDRESS OF ERROR REG
027320  012701  000001                    MOV     #BIT0,R1        ;R1 WILL HAVE BIT UNDER TEST
027324  052777  000040  152274    2$:     BIS     #CLR,@RHCS2     ;CLEAR
027332  013777  001760  152266            MOV     @#UNIT,@RHCS2   ;REINSTATE UNIT NO.
027340  010112                            MOV     R1,@R2          ;SET ERROR BIT
027342  004737  037470                    JSR     PC,@#PUTREG     ;SAVE REGISTERS
027346  120537  001716                    CMPB    R5,@#AS         ;ONLY THE BIT IN R5 SHOULD BE
                                                                  ;SET IN RHAS
027352  001401                            BEQ     3$              ;BRANCH IF GOOD
027354  104020                            ERROR   20              ;WITH THE SETTING OF ONE
                                                                  ;ERROR BIT IN A ERROR
                                                                  ;REGISTER THE CORRESPONDING
                                                                  ;RHAS BIT DID NOT SET
027356  013746  001722            3$:     MOV     @#DS1,-(SP)     ;GET RHDS1
027362  042716  001100                    BIC     #VV!PROG,(SP)   ;REMOVE VV AND PROG
027366  022726  140600                    CMP     #ATA!ERR!DPR!DRY,(SP)+;THISE BITS PLUS VV SHOULD BE IN RHDS1
027372  001401                            BEQ     4$              ;BRANCH IF GOOD
027374  104020                            ERROR   20              ;WITH THE SETTING OF ONE
                                                                  ;ERROR BIT COMPOSIT ERROR
                                                                  ;AND ATTENTION ACTIVE
                                                                  ;TOGETHER WITH THE OTHER
                                                                  ;PERMANENT BITS DID NOT SET

027376  012777  000001  152224    4$:     MOV     #GO,@RHCS1      ;GIVE NO-OP
```

```
7550  027404  004737  037470                  JSR     PC,@#PUTREG     ;SAVE REGISTERS
7551  027410  020112                          CMP     R1,@R2  ;GO SHOULD NOT CLEAR ERROR
7552  027412  001410                          BEQ     5$
7553  027414  010237  037534                  MOV     R2,@#REGADR     ;FAILING REGISTER
7554  027420  010137  001124                  MOV     R1,@#$GDDAT     ;GOOD DATA
7555  027424  013737  001676  001126          MOV     @#CS2,@#$BDDAT  ;BAD DATA
7556  027432  104001                          ERROR   1               ;"GO" WITH NO-OP CHANGED
7557                                                                  ;ERROR REGISTER
7558  027434  013746  001722          5$:     MOV     @#DS1,-(SP)     ;GET RHDS1
7559  027440  042716  001100                  BIC     #VV!PROG,(SP)   ;CLEAR VV AND PROG
7560  027444  022726  140600                  CMP     #ATA!ERR!DPR!DRY,(SP)+;GO SHOULD NOT CLEAR ANY BITS
7561  027450  001401                          BEQ     7$
7562  027452  104020                          ERROR   20              ;"GO" WITH NO-OP SHOULD NOT CLEAR
7563                                                                  ;ATA AND/OR ERR
7564  027454  006301          7$:     ASL     R1              ;GET BIT ONE LEFT
7565  027456  103403                  BCS     10$             ;BRANCH IF 16 BITS DONE
7566  027460  031001                  BIT     (R0),R1         ;IS THIS BIT TO BE TESTED
7567  027462  001374                  BNE     7$              ;IF NOT TO BE TESTED BRANCH
7568  027464  000717                  BR      2$              ;IF TO BE TESTED BRANCH
7569  027466  005720          10$:    TST     (R0)+           ;ADVANCE R0 TO NEXT ERROR REG.
7570  027470  005337  001200          DEC     @#$TMP1         ;COUNT 3 REGISTERS
7571  027474  001310                  BNE     1$              ;BRANCH IF 3 NOT COMPLETE
7572
7573                                  ;NOW AFTER SETTING ATA IN RHDS1 "1" IN RHAS AT THE
7574                                  ;DRIVE POSITION SHOULD CLEAR ATA IN RHDS1
7575
7576  027476  004737  040064  11$:     JSR     PC,@#CLDISK     ;CLEAR
7577  027502  012767  027476  151400  MOV     #11$,$LPERR     ;ERROR RETURN
7578  027510  012714  177777          MOV     #-1,@R4         ;SET BIT IN RHAS AND ATA IN RHDS1
7579  027514  013777  002002  152124  MOV     @#ATTENT,@RHAS  ;WRITE 1 INTO  DRIVE BIT POSITION
7580  027522  004737  037470          JSR     PC,@#PUTREG     ;SAVE REGISTERS
7581  027526  105737  001716          TSTB    @#AS            ;THIS SHOULD BE ZERO
7582  027532  001401                  BEQ     12$
7583  027534  104020                  ERROR   20              ;MOVING A "1" INTO RHAS
7584                                                          ;AT THE DRIVE BIT POSITION
7585                                                          ;DID NOT CLEAR IT
7586  027536  013746  001722          12$:    MOV     @#DS1,-(SP)     ;GET RHDS1
7587  027542  042716  001000          BIC     #PROG,(SP)      ;MASK PROGRAMABLE
7588  027546  022726  040700          CMP     #ERR!VV!DPR!DRY,(SP)+
7589                                                          ;RHDS1 SHOULD HAVE THESE BITS
7590                                                          ;BUT ATA SHOULD BE CLEARED
7591  027552  001401                  BEQ     13$             ;BRANCH IF GOOD
7592
7593  027554  104020                  ERROR   20              ;MOVING "1" INTO RHAS AT THE
7594                                                          ;DRIVE BIT POSITION DID NOT
7595                                                          ;CLEAR ATA IN RHDS1
7596  027556  022737  177777  001702  13$:    CMP     #-1,@#ER1               ;RHER1 SHOULD NOT CHANGE
7597                                  ;BY CLEARING RHAS
7598
7599  027564  001401                  BEQ     TST67   ;BRANCH IF GOOD
7600
7601  027566  104020                  ERROR   20              ;RHER1 CHANGED BU CLEARING
7602                                                          ;RHAS BY MOVING "1" INTO
7603                                                          ;THE DRIVE BIT POSITION
7604
7605
```

```
                                ;;*****************************************************************
                                ;*TEST 67       BUS ADDRESS INHIBIT

                                ;*      READ CYLINDER0, FORMAT 16 BITS PER WORD
                                ;*      TRACK0, SECTOR 1, KEYS 0, 10 WORDS OF 177400
                                ;*      THIS IS DONE WITH BUS ADDRESS INHIBIT SET
                                ;*      ANY ERROR LOGIC INDICATION IS NOT CONCLUSIVE ON FIRST PASS
                                ;*      BECAUSE ERROR LOGIC HAS NOT BEEN CHECKED YET (ON FIRST PASS)

                                ;;*****************************************************************
027570  000004                  TST67:  SCOPE
027572  012706  001000                  MOV     #STACK,SP       ;RESET STACK

027576  012737  000067  004172          MOV     #TTNO,@#TSTNM   ;THIS SAVES TEST NUMBER

027604  004037  040002                  JSR     R0,@#CLAREA     ;CLEAR SIMULATED DISK
027610  046330                          .WORD   DISK            ;FROM
027612  047326                          .WORD   DISK+776        ;TO
027614  177400                          .WORD   177400          ;DATA

027616  004037  040002                  JSR     R0,@#CLAREA     ;CLEAR READ INTO BUFFER
027622  003126                          .WORD   REINTO          ;FROM
027624  004124                          .WORD   REINTO+776      ;TO
027626  000000                          .WORD   0               ;DATA

                                ;THESE ARE TO SETUP FOR DISKLESS USE ONLY

027630  012737  010000  044412          MOV     #FMT22,@#CYL    ;CYLINDER 0 16 BITS PER WORD FORMAT
027636  105037  044415                  CLRB    @#SECOTR+1      ;TRACK 0
027642  112737  000001  044414          MOVB    #1,@#SECOTR     ;SECTOR 1
027650  005037  044416                  CLR     @#KEY1          ;KEY1=0
027654  005037  044420                  CLR     @#KEY2          ;KEY2=0
027660  012737  000012  044472          MOV     #10.,@#DAWORD   ;NO. OF DATA WORDS
027666  005037  044422                  CLR     @#X             ;THIS IS A READ COMMAND
027672  004537  041242                  JSR     R5,@#CRC        ;GO TO CALCULATE CRC
027676  044412                          CYL
027700  046312                          WCRC

                                ;THESE ARE REGULAR SETUPS

027702  004737  040064                  JSR     PC,@#CLDISK     ;SETUP GENERAL REGISTERS
027706  013711  002042                  MOV     @#READAT,@R1    ;READ DATA INTO RHCS1=70
027712  012777  177766  151702          MOV     #-10.,@RHWC     ;10 DATA WORDS
027720  012777  003126  151676          MOV     #REINTO,@RHBA   ;STARTING ADDRESS OF READ BUFFER
027726  112746  000001                  MOVB    #1,-(SP)        ;IN LOWER BYTE GET SECTOR 1
027732  112766  000000  000001          MOVB    #0,1(SP)        ;GET TRACK0 IN UPPER BYTE
027740  012677  151670                  MOV     (SP)+,@RHDST    ;TRACK/SECTOR IN RHDST
027744  012777  014000  151666          MOV     #FMT22!ECI,@RHOF ;16 BITS PER WORD
                                                                ;ECC CORRECTION INHIBIT BECAUSE
                                                                ;ECC IS NOT CHECKED HERE
027752  005077  151664                  CLR     @RHCA           ;CYLINDER 0
```

```
7662  027756  004737  040120                  JSR     PC,@#CHECKT     ;CHECK FOR DVA,RDY,MOL,DPR,DRY
7663  027762  052777  000010  151636          BIS     #BAI,@RHCS2     ;SET BUS ADDRESS INHIBIT
7664  027770  005037  001772                  CLR     @#ERFLG$        ;CLEAR ERROR FLAG
7665  027774  004737  044302                  JSR     PC,@#COMHD      ;READ DATA
7666
7667                                          ;IF THE PROGRAM COMES BACK HERE WITHOUT ERROR PRINTOUT
7668                                          ;FROM "COMHD" ROUTINE IN MEANS DATA IS TO BE CHECKED
7669
7670                                          ;NOW THE DATA READ INTO "REINTO" BUFFER WILL
7671                                          ;BE CHECKED, ONLY ONE WORD SHOULD BE CHANGED
7672                                          ;ALL OTHER WORDS SHOULD REMAIN UNCHANGED
7673                                          ;THE "WRFROM" BUFFER IS FILLED WITH EXPECTED DATA AND CHECKED
7674
7675  030000  005037  001772                  CLR     @#ERFLG$        ;CLEAR FLAG
7676  030004  004037  040002                  JSR     R0,@#CLAREA     ;CLEAR BUFFER
7677  030010  002062                          WRFROM                  ;FROM
7678  030012  003060                          WRFROM+776              ;TO
7679  030014  000000                          0                       ;DATA
7680
7681                                          ;EXPECTED DATA IS 177400 IN FIRST LOCATION ONLY
7682  030016  012737  177400  002062          MOV     #177400,@#WRFROM        ;EXPECTED DATA
7683
7684                                          ;NOW READ DATA BUFFER IS CHECKED
7685
7686  030024  012700  002062                  MOV     #WRFROM,R0      ;GOOD DATA
7687  030030  012701  003126                  MOV     #REINTO,R1      ;DATA READ
7688  030034  012702  000400                  MOV     #256.,R2        ;COUNTER
7689  030040  012737  000401  044532  1$:     MOV     #257.,@#ERWORD  ;FOR ERROR WORD NO
7690  030046  022021                          CMP     (R0)+,(R1)+     ;COMPARE GOOD WITH READ BUFFER
7691  030050  001424                          BEQ     2$              ;BRANCH IF GOOD
7692  030052  014037  001124                  MOV     -(R0),@#$GDDAT  ;GOOD DATA
7693  030056  014137  001126                  MOV     -(R1),@#$BDDAT  ;BAD DATA
7694  030062  160237  044532                  SUB     R2,@#ERWORD     ;ERROR WORD NO
7695  030066  005737  001772                  TST     @#ERFLG$                ;ANY ERRORS ALREADY THERE
7696  030072  001002                          BNE     3$              ;IF YES BRANCH DO NOT TYPE HEADER
7697  030074  104004                          ERROR   4               ;ERROR ON READ DATA
7698  030076  000401                          BR      4$              ;BRANCH TO AVOID PRINTING NEXT ERROR
7699  030100  104005                  3$:     ERROR   5               ;WORD NO 1-10 ARE DATA
7700                                                                  ;WORDS
7701                                                                  ;WORD NOS 11-256 HAVE NOT BEEN
7702                                                                  ;READ AND BUFFER SHOULD BE
7703                                                                  ;ZERO IF OTHER THAN ZERO
7704                                                                  ;WRONG NUMBER OF WORDS HAVE
7705                                                                  ;BEEN READ IN THE DISK NOW
7706                                                                  ;CONTAINS 177400 ALL 256
7707                                                                  ;WORDS BUT ONLY 10 WORDS
7708                                                                  ;SHOULD BE READ IN
7709
7710  030102  022021                  4$:     CMP     (R0)+,(R1)+     ;UNDO -(R0) AND -(R1) FOR ERROR
7711  030104  013746  001140                  MOV     @#SWR,-(SP)     ;GET SWITCH SETTING
7712  030110  042716  177177                  BIC     #177177,(SP)    ;KEEP ONLY SWITCH 7 AND 8
7713  030114  022726  000200                  CMP     #SW07,(SP)+     ;IS 7 SET AND 8 RESET
7714
7715  030120  001402                          BEQ     TST70   ;BRANCH OUT IF YES
7716
7717  030122  005302                  2$:     DEC     R2              ;COUNT
```

```
7718  030124  001345                        BNE     1$              ;BRANCH IF NOT COMPLETE
7719
7720
7721
7722
7723
7724
7725                                ;;*****************************************************************
7726                                ;*TEST 70       RHCS2 - BIT # 11 - NEM
7727
7728                                ;*      READ CYLINDER0, FORMAT 16 BITS PER WORD
7729                                ;*      TRACK0, SECTOR 1, KEYS 0, 1 WORD OF 177400
7730                                ;*      THIS IS DONE WITH BUS ADDRESS INHIBIT SET
7731                                ;*      BUS ADDRESS USED IS 760000 THIS IS ALWAYS NON EXISTANT
7732                                ;*      THIS SHOULD SET NEM
7733
7734                                ;;*****************************************************************
7735  030126  000004                TST70:  SCOPE
7736  030130  012706  001000                MOV     #STACK,SP       ;RESET STACK
7737
7738
7739  030134  012737  000070  004172        MOV     #TTNO,@#TSTNM   ;THIS SAVES TEST NUMBER
7740
7741  030142  000167  000310                JMP     TST71   ;JUMP TO NEXT TEST FOR RH70
7742  030146  004037  040002                JSR     R0,@#CLAREA     ;CLEAR SIMULATED DISK
7743  030152  046330                        .WORD   DISK            ;FROM
7744  030154  047326                        .WORD   DISK+776        ;TO
7745  030156  177400                        .WORD   177400          ;DATA
7746
7747
7748                                ;THESE ARE TO SETUP FOR DISKLESS USE ONLY
7749
7750  030160  012737  010000  044412        MOV     #FMT22,@#CYL    ;CYLINDER 0 16 BITS PER WORD FORMAT
7751  030166  105037  044415                CLRB    @#SECOTR+1      ;TRACK 0
7752  030172  112737  000001  044414        MOVB    #1,@#SECOTR     ;SECTOR 1
7753  030200  005037  044416                CLR     @#KEY1          ;KEY1=0
7754  030204  005037  044420                CLR     @#KEY2          ;KEY2=0
7755  030210  012737  000001  044472        MOV     #1.,@#DAWORD    ;NO. OF DATA WORDS
7756  030216  005037  044422                CLR     @#X             ;THIS IS A READ COMMAND
7757  030222  004537  041242                JSR     R5,@#CRC        ;GO TO CALCULATE CRC
7758  030226  044412                        CYL
7759  030230  046312                        WCRC
7760
7761                                ;THESE ARE REGULAR SETUPS
7762
7763  030232  004737  040064                JSR     PC,@#CLDISK     ;SETUP GENERAL REGISTERS
7764  030236  013711  002042                MOV     @#READAT,@R1    ;READ DATA INTO RHCS1=70
7765  030242  012777  177777  151352        MOV     #-1.,@RHWC      ;10 DATA WORDS
7766  030250  012777  160000  151346        MOV     #160000,@RHBA   ;STARTING ADDRESS OF READ BUFFER
7767  030256  052711  001400                BIS     #A16!A17,@R1    ;IS 760000
7768  030262  112746  000001                MOVB    #1,-(SP)        ;IN LOWER BYTE GET SECTOR 1
7769  030266  112766  000000  000001        MOVB    #0,1(SP)        ;GET TRACK0 IN UPPER BYTE
7770  030274  012677  151334                MOV     (SP)+,@RHDST    ;TRACK/SECTOR IN RHDST
7771  030300  012777  014000  151332        MOV     #FMT22!ECI,@RHOF ;16 BITS PER WORD
7772                                                                ;ECC CORRECTION INHIBIT BECAUSE
7773                                                                ;ECC IS NOT CHECKED HERE
```

```
7774  030306  005077  151330                CLR     @RHCA           ;CYLINDER 0
7775  030312  004737  040120                JSR     PC,@#CHECKT     ;CHECK FOR DVA,RDY,MOL,DPR,DRY
7776  030316  052777  000010  151302        BIS     #BAI,@RHCS2     ;SET BUS ADDRESS INHIBIT
7777  030324  005037  001772                CLR     @#ERFLG$        ;CLEAR ERROR FLAG
7778  030330  004737  044302                JSR     PC,@#COMHD      ;READ DATA
7779
7780
7781
7782  030334  011137  001126        1$:     MOV     @R1,@#$BDDAT    ;TEST DATA
7783
7784  030340  022737  145670  001126        CMP     #SC!TRE!DVA!A16!A17!RDY!7D,@#$BDDAT ;COMPARE RHCS1
7785  030346  001406                        BEQ     2$              ;BRANCH IF GOOD
7786  030350  012737  144270  001124        MOV     #SC!TRE!DVA!RDY!7D,@#$GDDAT ;GOOD DATA
7787  030356  010137  037534                MOV     R1,@#REGADR     ;REGISTER RHCS1
7788  030362  104001                        ERROR   1               ;REFERENCE NON EXISTANT
7789                                                                ;MEMORY DID NOT SET
7790                                                                ;REQUIRED BITS
7791  030364  013746  001760        2$:     MOV     @#UNIT,-(SP)    ;GET UNIT NUMBER
7792  030370  052716  004110                BIS     #NEM!IR!BAI,(SP)        ;INCLUDE NEM BAI AND IR
7793  030374  012637  001124                MOV     (SP)+,@#$GDDAT  ;
7794  030400  011237  001126                MOV     @R2,@#$BDDAT    ;TEST DATA
7795  030404  023737  001124  001126        CMP     @#$GDDAT,@#$BDDAT;COMPARE RHCS2
7796  030412  001403                        BEQ     3$
7797  030414  010237  037534                MOV     R2,@#REGADR     ;REGISTER ADDRESS
7798  030420  104001                        ERROR   1               ;REFRENCING NONEXISTANT MEMORY
7799                                                                ;CAUSED AN ERROR SHOULD SET NEM
7800  030422  017737  151176  001126  3$:   MOV     @RHBA,@#$BDDAT  ;TEST DATA
7801
7802  030430  022737  160000  001126        CMP     #160000,@#$BDDAT        ;COMPARE RHBA
7803  030436  001407                        BEQ     4$              ;BRANCH IF GOOD
7804  030440  012737  160000  001124        MOV     #160000,@#$GDDAT;GOOD DATA
7805  030446  013737  001624  037534        MOV     @#RHBA,@#REGADR ;REGISTER ADDRESS RHBA
7806  030454  104001                        ERROR   1               ;AFTER A NON EXISTANT MEMORY ERROR
7807                                                                ;RHBA DOES NOT HAVE 160002
7808  030456                        4$:
7809
7810
7811
7812
7813
7814
7815
7816
7817                                ;;*****************************************************************
7818                                ;*TEST 71       WRITE CHECK ERROR
7819
7820                                ;*      WRITE CHECK DATA CYLINDER 0 FORMAT 16 BITS PER WORD
7821                                ;*      TRACK 1, SECTOR 1, KEYS 0, 32 WORDS OF DATA
7822                                ;*      FIFTH WORD IS CHANGED ON DISK TO GIVE WRITE CHECK ERROR
7823                                ;*      ANY DEVICE LOGIC ERROR INDICATIONS ARE NOT CONCLUSIVE
7824                                ;*      ON FIRST PASS
7825                                ;*      BECAUSE ERROR LOGIC HAS NOT YET BEEN CHECKED
7826                                ;*      ONLY RH WRITE CHECK ERROR IS TESTED
7827
7828                                ;;*****************************************************************
7829  030456  000004                TST71:  SCOPE
```

```
                                        ;DATA TABLE
                                        ;TOTAL OF 32 WORDS CONSISTING OF
                                        ;16 WORDS OF FLOATING ONES (EG. 1, 2, 4, 10)
                                        ;16 WORDS OF FLOATING ZEROS (EG. 177776, 177775)

030460  012706  001000                  MOV     #STACK,SP       ;RESET STACK

030464  012737  000071  004172          MOV     #TTNO,@#TSTNM   ;THIS SAVES TEST NUMBER

030472  004737  040064                  JSR     PC,@#CLDISK     ;INIT AND SET GENERAL REGISTERS
                                        ;SET UP "REINTO" FOR WHAT IS TO BE READ

030476  012700  000001                  MOV     #1,R0           ;GETTING READY TO FLOAT 1
030502  012701  003126                  MOV     #REINTO,R1      ;STARTING ADDRESS WHERE 1 GOES
030506  010021          1$:             MOV     R0,(R1)+        ;MOVE FLOATING 1
030510  006100                          ROL     R0              ;GET 1 ONE BIT LEFT
030512  103375                          BCC     1$              ;BRANCH IF 16 NOT DONE
030514  012700  177776                  MOV     #177776,R0      ;GETTING READY TO FLOAT 0
030520  012701  003166                  MOV     #REINTO+<16.*2>,R1 ;STARTING ADDRESS WHERE 177776 GOES
030524  010021          2$:             MOV     R0,(R1)+        ;MOVE IN FLOATING 0
030526  000261                          SEC                     ;SET CARRY
030530  006100                          ROL     R0              ;GET 0 ONE BIT LEFT
030532  103774                          BCS     2$              ;BRANCH IF 16 NOT DONE

030534  004037  040002                  JSR     R0,@#CLAREA     ;FILL REST OF BUFFER WITH 1
030540  003226                          .WORD   REINTO+<32.*2>  ;FROM
030542  004124                          .WORD   REINTO+776      ;TO
030544  000001                          .WORD   1               ;WITH DATA

                                        ;SET UP SIMULATED DISK WITH WHAT IS TO BE READ

030546  012700  000001                  MOV     #1,R0           ;GETTING READY TO FLOAT 1
030552  012701  046330                  MOV     #DISK,R1        ;STARTING ADDRESS WHERE 1 GOES
030556  010021          3$:             MOV     R0,(R1)+        ;MOVE FLOATING 1
030560  006100                          ROL     R0              ;GET 1 ONE BIT LEFT
030562  103375                          BCC     3$              ;BRANCH IF 16 NOT DONE

030564  012700  177776                  MOV     #177776,R0      ;GETTING READY TO FLOAT 0
030570  012701  046370                  MOV     #DISK+<16.*2>,R1 ;STARTING ADDRESS WHERE 177776 GOES
030574  010021          4$:             MOV     R0,(R1)+        ;MOVE FLOATING 0
030576  000261                          SEC                     ;SET CARRY
030600  006100                          ROL     R0              ;GET 0 ONE BIT LEFT
030602  103774                          BCS     4$              ;BRANCH IF 16 NOT DONE

030604  004037  040002                  JSR     R0,@#CLAREA     ;FILL REST OF BUFFER WITH 0
030610  046430                          .WORD   DISK+<32.*2>    ;FROM
030612  047326                          .WORD   DISK+776        ;TO
030614  000000                          .WORD   0               ;WITH DATA

                                        ;CHANGE FIFTH WORD TO 0 ON DISK
030616  005037  046340                  CLR     @#DISK+10       ;CLEAR FIFTH WORD ON DISK
030622  005037  001772                  CLR     @#ERFLG$        ;CLEAR ERROR FLAG
030626  004737  041100                  JSR     PC,@#WRCHDA     ;WRITE CHECK DATA
```

```
                                                        ;CYLINDER 0, TRACK 1, SECTOR 1
                                                        ;KEYS 0, 32 WORDS.

                                                ;IF THE PROGRAM COMES BACK HERE THEN WRITE CHECK
                                                ;HAS BEEN COMPLETED NOW WRITE CHECK ERROR BIT IS TESTED

030632  013746  001760                MOV     @#UNIT,-(SP)    ;GET UNIT NUMBER
030636  052716  040300                BIS     #40300,(SP)     ; BIT 6+7 SHOULD BE SET
030642  004737  037470                JSR     PC,@#PUTREG     ;SAVE REGISTERS
030646  022637  001676                CMP     (SP)+,@#CS2     ;COMPARE RHCS2
030652  001407                        BEQ     6$              ;BRANCH IF GOOD
030654  032737  040000  001676        BIT     #WCE,@#CS2      ;WRITE CHECK ERROR HIGH?
030662  001002                        BNE     5$              ;BRANCH IF ERROR NOT DUE TO "WCE"
030664  104017                        ERROR   17              ;RHDB CONTAINS FAILING WORD
030666  000401                        BR      6$              ;RHBA CONTAINS ADDRESS+2
                                                        ;OF THE WORD IN MEMORY FROM
                                                        ;THE DISK THAT DID NOT COMPARE
                                                        ;TRE AND SC WILL BE SET DUE TO WCE
030670  104017          5$:           ERROR   17              ;WCE WAS CORRECTLY NOT SET
                                                        ;BUT SOME BITS OTHER THAN
                                                        ;IR AND UNIT NO. WERE SET

030672  022737  177750  001672  6$:   CMP     #-24.,@#WC      ;COMPARE RHWC AFTER A FORCED
                                                        ;WRITE CHEECK ERROR
030700  001402                        BEQ     14$             ;BRANCH IF GOOD
030702  104017                        ERROR   17              ;WORD COUNT REGISTER IN ERROR AFTER A
                                                        ;FORCED WRITECHECK ERROR ON FIFTH WORD
030704  000405                        BR      15$             ;BRANCH TO CONTINUE
030706  022737  003146  001674  14$:  CMP     #REINTO+<8.*2>,@#BA;COMPARE RHBA AFTER FORCED
                                                        ;WRITECHECK ERROR IN FIFTH WORD
030714  001401                        BEQ     15$             ;BRANCH IF GOOD
030716  104017                        ERROR   17              ;BUS ADDRESS REGISTER IN ERROR AFTER
                                                        ;FORCED WRITE CHECK ERROR ON FIFTH WORD
                                                ;NOW CHECK MEMORY TO SEE IF ANYTHING GOT DESTROYED
                                                ;FILL "WRFROM" WITH WHAT SHOULD BE IN REINTO THEN CHECK

030720  005037  001772          15$:  CLR     @#ERFLG$        ;CLEAR ERROR FLAG
030724  012700  000001                MOV     #1,R0           ;GETTING READY TO FLOAT 1
030730  012701  002062                MOV     #WRFROM,R1      ;START ADDRESS WHERE 1 GOES
030734  010021                  7$:   MOV     R0,(R1)+        ;MOVE FLOATING 1
030736  006100                        ROL     R0              ;GET 1 ONE BIT LEFT
030740  103375                        BCC     7$              ;BRANCH IF 16 NOT DONE

030742  012700  177776                MOV     #177776,R0      ;GETTING READY TO FLOAT 0
030746  012701  002122                MOV     #WRFROM+<16.*2>,R1 ;STARTING ADDRESS WHERE 177776 GOES
030752  010021                  10$:  MOV     R0,(R1)+        ;MOVE IN FLOATING 0
030754  000261                        SEC                     ;SET CARRY
030756  006100                        ROL     R0              ;GET 0 ONE BIT LEFT
030760  103774                        BCS     10$             ;BRANCH IF CARRY SET

030762  004037  040002                JSR     R0,@#CLAREA     ;FILL REST OF BUFFER WITH 1
030766  002162                        .WORD   WRFROM+<32.*2>  ;FROM
030770  003060                        .WORD   WRFROM+776      ;TO
030772  000001                        .WORD   1               ;WITH DATA

                                                ;NOW THE READ BUFFER WILL BE CHECKED
```

```
                                                ;CHECK
031000  002062                                  ;GOOD BUFFER
031002  003126                                  ;TEST BUFFER
031004  000400                                  ;NUMBER OF WORDS CHECKED
031006  031014                                  ;RETURN POINT FOR ERROR HEADER
031010  031020                                  ;RETURN POINT FOR ERROR DATA

031012  031026                                  ;RETURN FOR GOOD COMPARISON

031014  104004          11$:    ERROR   4       ;READ NEXT ERROR 5
031016  000207                  RTS     PC      ;RETURN TO COMPARISON SUBROUTINE
031020  104005          12$:    ERROR   5       ;DATA IN REINTO BUFFER GOT
                                                ;CHANGED AFTER A WRITE
                                                ;CHECK DATA COMMAND
                                                ;WORD NO CONTAINS THE WORD
                                                ;NUMBER THAT GOT CHANGED
031022  000207                  RTS     PC      ;RETURN TO COMPARISON SUBROUTINE

031024  000240          13$:    NOP             ;ONLY A BRANCH POINT



                        ;;*****************************************************************
                        ;*TEST 72       ERROR REGISTER #1-BIT 4 -FORMAT ERROR
                        ;*      THE SIMULATED DISK IS FILLED WITH CYLINGER 0 TRACK 1
                        ;*      SECTOR 0 FORMAT=18 BITS PER WORD AND 4 WORDS
                        ;*      OF 125252, A READ HEADER AND DATA COMMAND IS GIVEN WITH 16 BITS
                        ;*      PER WORD FORMAT, FER=BIT4 SHOULD SET BUT THE
                        ;*      READ SHOULD BE COMPLETE

                        ;;*****************************************************************
031026  000004          TST72:  SCOPE


031030  012706  001000          MOV     #STACK,SP       ;RESET STACK

031034  012737  000072  004172  MOV     #TTNO,@#TSTNM   ;THIS SAVES TEST NUMBER


                        ;       SETUP FOR WHAT IS TO BE READ
                        ;       HEADER CRC IS RESTORED FROM A SUBROUTINE

031042  012746  125252          MOV     #125252,        -(SP)   ;DATA TO BE READ
031046  012705  000400          MOV     #256.,  R5      ;COUNTER
031052  012700  046330          MOV     #DISK,  R0      ;START OF SIMULATED DISK DATA
031056  011620          1$:     MOV     (SP),   (R0)+   ;MOVE IN DATA ON TO SIMULATED DISK
031060  005305                  DEC     R5              ;COUNT
031062  001375                  BNE     1$              ;BRANCH IF 256 NOT COMPLETE
031064  005726                  TST     (SP)+           ;UNDO -(SP)
031066  012705  000021          MOV     #17.,   R5      ;2 ECC WORDS
```

```
                                                        ;1 DATA GAP
                                                        ;14 TOLERANCE GAP
031072  005020              2$:     CLR     (R0)+       ;CLEAR ECC, DATA GAP, AND
031074  005305                      DEC     R5          ;TOLERANCE GAP
031076  001375                      BNE     2$          ;BRANCH IF NOT COMPLETE

                                    ;THESE ARE TO SETUP FOR DISKLESS USE ONLY

031100  012737  000000  044412      MOV     #0!0,   @#CYL   ;16 BITS PER WORD
                                                            ;CYLINDER 0, FORMAT 16 BITS
031106  112737  000001  044415      MOVB    #1,     @#SECOTR+1 ;TRACK 1
031114  112737  000000  044414      MOVB    #0,     @#SECOTR ;SECTOR 0
031122  012737  000000  044416      MOV     #0,     @#KEY1  ;KEY1=0
031130  012737  000000  044420      MOV     #0,     @#KEY2  ;KEY2=0
031136  012737  000004  044472      MOV     #4.,    @#DAWORD ;NO. OF DATA WORDS
031144  005037  044422              CLR     @#X             ;THIS IS A READ COMMAND
031150  004537  041242              JSR     R5,@#CRC        ;GO TO CALCULATE CRC
031154  044412                      CYL
031156  046312                      WCRC

                                    ;THESE ARE REGULAR SETUPS
031160  004737  040064              JSR     PC,@#CLDISK     ;SETUP GENERAL REGISTERS
031164  012777  177770  150430      MOV     #-4.-4.,@RHWC   ;4. DATA 4 HEADER WORDS
031172  012777  003126  150424      MOV     #REINTO,@RHBA   ;STARTING ADDRESS OF READ BUFFER
031200  112746  000000              MOVB    #0,     -(SP)   ;IN LOWER BYTE GET SECTOR
031204  112766  000001  000001      MOVB    #1,     1(SP)   ;GET TRACK IN HIGHER BYTE
031212  012677  150416              MOV     (SP)+,  @RHDST  ;TRACK/SECTOR IN RHDST
031216  012777  014000  150414      MOV     #FMT22!ECI,@RHOF ;16 BITS PER WORD
                                                        ;ECC CORRECTION INHIBIT
                                                        ;BECAUSE ECC IS NOT GOING
                                                        ;TO BE CHECKED
031224  005077  150412              CLR     @RHCA           ;CYLINDER 0

031230  004737  040120              JSR     PC,     @#CHECKT ;CHECK FOR DVA,RDY,MOL,DPR,DRY

031234  013711  002044              MOV     @#REFOR,@R1     ;READ HEADER AND DATA=72
031240  005037  001772              CLR     @#ERFLG$                ;CLEAR ERROR FLAG
031244  004737  044302              JSR     PC,     @#COMHD ;READ HEADER AND DATA

                                    ;IF THE PROGRAM COMES BACK HERE WITHOUT ERROR PRINTOUTS
                                    ;FROM THE "COMHD" ROUTINE THAT MEANS SECTOR GAP,
                                    ;FIRST SYNC, HEADER, HEADER CRC, HEADER GAP AND
                                    ;SYNC BYTE HAVE GONE BY AND SYNCS WERE CORRECTLY
                                    ;DETECTED
                                    ;HEADER AND DATA ARE TO BE CHECKED.
                                    ;IN CHECKING READ DATA THE WRITE FROM BUFFER
                                    ;"WRFROM" IS FILLED WITH EXPECTED DATA AND
                                    ;COMPARISONS ARE MADE

031250  005737  001772              TST     @#ERFLG$                ;ANY ERRORS ALREADY THERE

031254  001052                      BNE     TST73   ;BRANCH IF YES
```

```
8054  031256  004737  040302                    JSR     PC,@#CHECKE     ;CHECK DVA,DRY,RDY,DPR
8055  031262  012700  002062                    MOV     #WRFROM,R0      ;GETTING READY TO FILL EXPECTED DATA
8056  031266  012720  000000                    MOV     #0,(R0)+        ;CYLINDER 0
8057  031272  112746  000000                    MOVB    #0,     -(SP)   ;IN LOWER BYTE GET SECTOR
8058  031276  112766  000001  000001           MOVB    #1,     1(SP)   ;GET TRACK IN HIGHER BYTE
8059  031304  012620                            MOV     (SP)+,  (R0)+   ;GET TRACK/SECTOR IN BUFFER
8060  031306  012720  000000                    MOV     #0,     (R0)+   ;KEY1 IN BUFFER
8061  031312  012720  000000                    MOV     #0,     (R0)+   ;KEY2 IN BUFFER
8062  031316  012701  000400                    MOV     #256.,  R1      ;DATA WORD COUNTER
8063  031322  012702  125252                    MOV     #125252,R2      ;DATA
8064  031326  010220                    3$:     MOV     R2,     (R0)+   ;DATA INTO BUFFER
8065  031330  005301                            DEC     R1              ;COUNT
8066  031332  001375                            BNE     3$              ;BRANCH IF 256 NOT DONE
8067
8068                                            ;NOW READ DATA BUFFER WILL BE CHECKED
8069
8070  031334  004037  040736                    JSR     R0,@#COMPAR     ;CHECK
8071  031340  002062                            WRFROM                  ;GOOD BUFFER
8072  031342  003126                            REINTO                  ;TEST BUFFER
8073  031344  000010                            4+4.                    ;NUMBER OF WORDS CHECKED
8074  031346  031354                            4$                      ;RETURN POINT FOR ERROR HEADER
8075  031350  031360                            5$                      ;RETURN POINT FOR ERROR DATA
8076  031352  031364                            6$                      ;RETURN FOR GOOD COMPARISON
8077  031354  104004                    4$:     ERROR   4               ;READ NEXT ERROR
8078  031356  000207                            RTS     PC              ;RETURN TO "COMPAR"
8079  031360  104005                    5$:     ERROR   5               ;WORD NOS 1 TO 4 ARE
8080                                                                    ;HEADER WORDS
8081                                                                    ;5 TO 260 ARE DATA WORDS
8082  031362  000207                            RTS     PC              ;RETURN TO "COMPAR"
8083
8084
8085
8086
8087                                            ;NOW SEE THAT FORMAT ERROR BIT GOT SET
8088
8089  031364  004737  037470            6$:     JSR     PC,@#PUTREG     ;SAVE REGISTERS
8090
8091  031370  022737  100020  001702            CMP     #FER!DCK,@#ER1  ;FORMAT ERROR SHOULD BE SET
8092
8093  031376  001401                            BEQ     TST73   ;BRANCH IF GOOD
8094
8095  031400  104020                            ERROR   20              ;A 16 BIT PER WORD READ WAS ATTEMPTED
8096                                                                    ;WHEN THE DISK HAD
8097                                                                    ;THE FORMAT BIT=0= 18 BITS PER
8098                                                                    ;WORD THE READ WAS
8099                                                                    ;COMPLETED BUT ERROR REG
8100                                                                    ;WAS NOT RIGHT
8101                                                                    ;NOTE DCK WILL BE SET BECAUSE
8102                                                                    ;ECC HAS NOT BEEN GENERATED
8103
8104
8105
8106                            ;;*********************************************************************
8107                            ;*TEST 73       ERROR REGISTER #1-BIT 4 -FORMAT ERROR
8108
8109                            ;*      THE SIMULATED DISK HEADER IS FILLED WITH CYLINDER 0
```

```
                                ;*      TRACK 0, SECTOR 0 FORMAT 18 BITS PER WORD
                                ;*      A WRITE DATA COMMAND IS GIVEN WITH SAME HEADER
                                ;*      EXCEPT FORMAT BIT.  THE DATA SHOULD NOT BE WRITTEN.
                                ;;*****************************************************************
031402  000004                 TST73:  SCOPE
                                        ;NOW A WRITE DATA WILL BE ATTEMPTED WITH
                                        ;WRONG FORMAT BIT
031404  012706  001000                 MOV     #STACK,SP       ;RESET STACK
031410  012737  000073  004172         MOV     #TTNO,@#TSTNM   ;THIS SAVES TEST NUMBER
031416  012737  177777  044526         MOV     #-1,@#NOSYNC    ;SET FLAG SO THAT DATA SYNC
                                                                ;AND DATA IS NOT READ
031424  004037  040002         FRMAT1: JSR     R0,@#CLAREA     ;CLEAR SIMULATED DISK
031430  046330                         .WORD   DISK            ;FROM
031432  047354                         .WORD   TOLGAP+16       ;TO
031434  000000                         .WORD   0               ;DATA
                                ;THESE ARE SETUP FOR DISKLESS USE ONLY
031436  005037  044412                 CLR     @#CYL           ;CYLINDER 0, FORMAT 18 BIT WORDS
031442  105037  044415                 CLRB    @#SECOTR+1      ;TRACK 0
031446  105037  044414                 CLRB    @#SECOTR        ;SECTOR 0
031452  005037  044416                 CLR     @#KEY1          ;KEY1 0
031456  005037  044420                 CLR     @#KEY2          ;KEY2 0
031462  012737  000004  044460         MOV     #4,@#NOWORD     ;NO OF DATA WORDS
031470  012737  000001  044422         MOV     #1,@#X          ;WRITE DATA
031476  004537  041242                 JSR     R5,@#CRC        ;GO TO CALCULATE CRC
031502  047516                         WCYL
031504  047526                         GCRC

                                ;THESE AER REGULAR SETUPS

031506  004037  040002                 JSR     R0,@#CLAREA     ;FILL WRITE FROM BUFFER WITH 125252
031512  002062                         WRFROM                  ;FROM
031514  002070                         WRFROM+6                ;TO
031516  125252                         125252                  ;DATA
031520  004737  040064                 JSR     PC,@#CLDISK     ;SETUP GENERAL REGISTERS
031524  012777  177774  150070         MOV     #-4,@RHWC       ;256 DATA WORDS
031532  012777  002062  150064         MOV     #WRFROM,@RHBA   ;STARTING ADDRESS OF WRITE BUFFER
031540  005077  150070                 CLR     @RHDST          ;TRACK=0 SECTOR=0
031544  012777  010000  150066         MOV     #FMT22,@RHOF    ;16 BITS PER WORD FORMAT
031552  005077  150064                 CLR     @RHCA           ;CYLINDER 0
031556  004737  040120                 JSR     PC,@#CHECKT     ;CHECK FOR DVA,RDY,MOL,DPR,DRY
031562  013711  002036                 MOV     @#WRIDAT,@R1    ;WRITE DATA=60
031566  005037  001772                 CLR     @#ERFLG$        ;CLEAR ERROR FLAG
031572  004737  044302                 JSR     PC,@#COMHD      ;WRITE DATA
                                ;IF THE PROGRAM COMES BACK HERE WITHOUT ERROR PRINTOUTS
                                ;FROM THE "COMHD" ROUTINE IT MEANS SECTOR GAP, SYNC BYTE
                                ;HEADER, HEADER CRC, HEADER GAP AND SYNC BYTE HAVE GONE BY
                                ;AND SYNCS WERE CORRECTLY DETECTED
                                ;DATA IS TO BE CHECKED
031576  004737  037470                 JSR     PC,@#PUTREG     ;SAVE REGISTERS
031602  005737  001772                 TST     @#ERFLG$        ;HAS ANY ERRORS OCCURED?
```

```
8166  031606  001041                          BNE     4$              ;BRANCH IF YES
8167  031610  012700  000000                  MOV     #0,R0           ;GOOD DATA
8168  031614  012701  046330                  MOV     #DISK,R1        ;DATA WRITTEN INTO "DISK"
8169  031620  012702  000004                  MOV     #4,R2           ;COUNTER
8170  031624  012737  000005  044532  1$:     MOV     #5,@#ERWORD     ;FOR ERROR WORD
8171  031632  020021                          CMP     R0,(R1)+        ;COMPARE GOOD DATA WITH DATA ON DISK
8172  031634  001424                          BEQ     3$              ;BRANCH IF GOOD
8173  031636  010037  001124                  MOV     R0,@#$GDDAT     ;GOOD DATA
8174  031642  014137  001126                  MOV     -(R1),@#$BDDAT  ;BAD DATA
8175  031646  160237  044532                  SUB     R2,@#ERWORD     ;ERROR WORD NO
8176  031652  005737  001772                  TST     @#ERFLG$        ;ANY ERRORS ALREADY THERE?
8177  031656  001002                          BNE     2$              ;BRANCH IF YES
8178  031660  104004                          ERROR   4               ;ERROR ON WRITE DATA COMMAND
8179                                                                  ;ON A WRITE DATA WITH
8180                                                                  ;WRONG FORMAT NO DATA
8181                                                                  ;SHOULD BE WRITTEN
8182                                                                  ;WORD NO GIVES WORD IN ERROR
8183  031662  000401                          BR      5$              ;BRANCH TO AVOID PRINTING NEXT ERROR
8184  031664  104005                  2$:     ERROR   5
8185  031666  005721                  5$:     TST     (R1)+           ;UNDO -(R1) FOR BAD DATA
8186  031670  013746  001140                  MOV     @#SWR,-(SP)     ;GET SWITCH SETTING
8187  031674  042716  177177                  BIC     #177177,(SP)    ;KEEP ONLY SWITCH 7 AND 8
8188  031700  022726  000200                  CMP     #SW07,(SP)+     ;IS 7 SET AND 8 RESET.
8189  031704  001402                          BEQ     4$              ;BRANCH IF YES
8190  031706  005302                  3$:     DEC     R2              ;IF NOT COUNT 256 WORDS
8191  031710  001345                          BNE     1$              ;BRANCH IF 256 NOT DONE
8192
8193                                          ;NOW CHECK TO SEE THAT FORMAT ERROR BIT GOT SET
8194
8195  031712  022737  000020  001702  4$:     CMP     #FER,@#ER1      ;FORMAT ERROR SHOULD BE SET
8196
8197  031720  001401                          BEQ     TST74   ;BRANCH IF GOOD
8198
8199  031722  104020                          ERROR   20              ;A 16 BIT PER WORD WRITE DATA
8200                                                                  ;WAS ATTEMPTED WHEN THE DISK
8201                                                                  ;HAD THE FORMAT BIT =0=18
8202                                                                  ;BITS PER WORD THE WRITE
8203                                                                  ;WAS CORRECTLY ABORTED
8204                                                                  ;BUT ERROR REG. 1 WAS WRONG
8205
8206
8207
8208                                  ;*      ERROR REGISTER #01 (RHER1) TEST
8209                                  ;*      BIT #1 (ILLEGAL REGISTER) CANNOT BE TESTED ON PDP11 THIS BIT
8210                                  ;*      IS FOR PDP10 USE ONLY
8211
8212
8213                                  ;;*****************************************************************
8214                                  ;*TEST 74       TEST ILF BIT #0 IN REG. RHER1
8215
8216                                  ;*      ILLEGAL FUNCTION SHOULD SET ATA,ERR,ILF
8217                                  ;*      A GO WITHOUT CLEARING ERR SHOULD SET MXF,DLT,TRE
8218
8219                                  ;;*****************************************************************
8220  031724  000004                  TST74:  SCOPE
8221
```

```
8222
8223  031726  012706  001000                  MOV     #STACK,SP       ;RESET STACK
8224
8225
8226  031732  012737  000074  004172          MOV     #TTNO,@#TSTNM   ;THIS SAVES TEST NUMBER
8227
8228
8229  031740  004737  040064                  JSR     PC,@#CLDISK     ;CLEAR REGISTERS
8230  031744  012777  000001  147676          MOV     #DMD,@RHMR      ;SET DIAGNOSTIC MODE
8231  031752  005037  002006                  CLR     @#TMPILL                ;GET READY TO MAKE ILLEGAL FUNCTION
8232  031756  012700  002016          1$:     MOV     #FUTABL,R0      ;GET READY TO MAKE ILLEGAL FUNCTION
8233  031762  012705  000021                  MOV     #17.,R5         ;COUNTER (16 GOOD FUNCTIONS)
8234  031766  023720  002006          2$:     CMP     @#TMPILL,(R0)+  ;IS THIS A LEGAL FUNCTION
8235  031772  001004                          BNE     3$              ;BRANCH IF NOT LEGAL
8236  031774  062737  000002  002006          ADD     #2,@#TMPILL     ;MAKE ANOTHER FUNCTION
8237  032002  000765                          BR      1$              ;GET READY TO TEST NEW FUNCTION
8238  032004  005305                  3$:     DEC     R5              ;NOT LEGAL SO DECREMENT COUNTER
8239  032006  001367                          BNE     2$              ;BRANCH IF 16 NOT COMPLETE
8240  032010  032737  000100  002006          BIT     #100,@#TMPILL   ;ALL BITS UP TO BIT #5 COMPARED?
8241  032016  001077                          BNE     12$             ;BRANCH IF 100
8242  032020  013737  002006  002060          MOV     @#TMPILL,@#ILLEGL;THIS IS AN ILLEGAL FUNCTION
8243  032026  062737  000002  002006          ADD     #2,@#TMPILL     ;GET READY TO TEST NEW FUNCTION NEXT TIME
8244  032034  004737  040064          4$:     JSR     PC,@#CLDISK
8245  032040  012777  000001  147602          MOV     #DMD,@RHMR      ;SET DIAGNOSTIC MODE
8246  032046  013711  002060                  MOV     @#ILLEGL,@R1    ;ILLEGAL FUNCTION
8247  032052  012737  032034  001110          MOV     #4$,@#$LPERR    ;ERROR RETURN POINT
8248  032060  004737  037470                  JSR     PC,@#PUTREG     ;SAVE REGISTERS
8249  032064  005737  001702                  TST     @#ER1           ;THERE SHOULD NOT BE ANY ERROR YET
8250  032070  001403                          BEQ     5$              ;BRANCH IF STILL ZERO
8251  032072  010437  001122                  MOV     R4,@#$BDADR     ;FAILING REGISTER ADDRESS RHER1
8252  032076  104011                          ERROR   11              ;ALTHOUGH AN ILLEGAL FUNCTION
8253                                                                  ;HAS BEEN MOVED INTO RHCS1
8254                                                                  ;NO ERRORS SHOULD SHOW TILL
8255                                                                  ;GO IS SET RHER1 SHOULD BE
8256                                                                  ;ALL ZEROS
8257  032100  052711  000001          5$:     BIS     #GO,@R1         ;GO IN RHCS1
8258  032104  004737  037470                  JSR     PC,@#PUTREG     ;SAVE REGISTERS
8259  032110  022737  000001  001702          CMP     #ILF,@#ER1      ;ILLEGAL FUNCTION SHOULD BE SET
8260  032116  001403                          BEQ     6$
8261  032120  010437  001122                  MOV     R4,@#$BDADR     ;FAILING REGISTER ADDRESS RHER1
8262  032124  104011                          ERROR   11              ;ILLEGAL FUNCTION DID NOT
8263                                                                  ;SET ON AN ILLEGAL FUNCTION
8264                                                                  ;EXECUTION, THE ILLEGAL FUNCTION
8265                                                                  ;BEING EXECUTED IS IN RHCS1
8266  032126  013746  001722          6$:     MOV     @#DS1,-(SP)     ;GET RHDS1
8267  032132  042716  001000                  BIC     #PROG,(SP)      ;MASK PROG
8268  032136  022726  140700                  CMP     #ATA!ERR!VV!DPR!DRY,(SP)+
8269                                                                          ;ATTENTION (BIT 15)
8270                                                                  ;VOLUME VALID (BIT 6)
8271                                                                          ;COMPOSIT ERROR (BIT 14)
8272                                                                  ;DEVICE READY (BIT 7) SHOULD
8273                                                                  ;BE SET ON RHDS1
8274  032142  001404                          BEQ     7$
8275  032144  013737  001652  001122          MOV     @#RHDS1,@#$BDADR        ;FAILING REGISTER ADDRESS RHDS1
8276  032152  104011                          ERROR   11              ;FOLLOWING BITS SHOULD BE SET
8277                                                                  ;WITH AN ILLEGAL FUNCTION
```

```
                                                                ;ATTENTION (BIT 15)
                                                                ;COMPOSIT ERROR (BIT 14)
                                                                ;MEDIUM ON LINE (BIT 12)
                                                                ;DEVICE READY (BIT 7)
032154  004737  042266          7$:     JSR     PC,@#MIDDLE     ;GIVE A WRITE HEADER AND
                                                                ;DATA COMMAND WITHOUT
                                                                ;CLEARING THE ERRORS
                                                                ;USING "MIDDLE" SO THAT
                                                                ;IT WILL COME BACK BEFORE
                                                                ;THE END TO FIND OUT ITS
                                                                ;STATE
032160  010237  032166                  MOV     R2,@#10$        ;MOVE RHCS2 ADDRESS
032164  104412                          WAT                     ;WAIT FOR MXF
032166  000000          10$:    .WORD   0                       ;ADDRESS OF RHCS2
032170  001000                  MXF
032172  004737  037470          11$:    JSR     PC,@#PUTREG     ;SAVE REGISTERS

032176  032737  040000  001700        BIT     #TRE,@#CS1      ;TRANSFER ERROR (BIT 14) RHCS1
                                                                ;SHOULD SET DUE TO MXF
032204  001003                  BNE     13$                     ;BRANCH IF GOOD
032206  010137  001122                  MOV     R1,@#$BDADR     ;FAILING REGISTER RHCS1
032212  104011                          ERROR   11              ;TRANSFER ERROR BIT 14 RHCS1
                                                                ;SHOULD BE SET DUE TO MXF
                                                                ;LOCAL SCOPE RETURN POINT
                                                                ;LOCAL SCOPE
032214  000660          13$:    BR      1$                      ;BRANCH FOR NEXT FUNCTION
032216  000240          12$:    NOP


                        ;;**************************************************************
                        ;*TEST 75       RHER1- BIT #2 - REG. MODIFICATION REFUSED

                        ;*      IN THIS TEST THE REGISTERS ARE IN TWO GROUPS
                        ;*      FIRST - RHCS1,RHDST,RHOF,RHCA,RHER1,RHER2,RHER3 - SETS RMR
                        ;*      SECOND - RHMR,RHAS - DOES NOT SET RMR
                        ;*      IF WRITING IS ATTEMPTED DURING AN OPERATION
                        ;*
                        ;*
                        ;*      ONLY ONE REGISTER IS WRITTEN INTO THAT IS RHCA
                        ;*
                        ;*
                        ;*      1 THE REGISTERS CONTENTS ARE SAVED IN "REINTO" BUFFER
                        ;*      2 WRITE HEADER AND DATA IS STARTED
                        ;*      3 ATTEMPT IS MADE TO WRITE INTO REGISTERS
                        ;*      4 ALL REGISTERS ARE COMPARED

                        ;;**************************************************************
032220  000004          TST75:  SCOPE


032222  012706  001000          MOV     #STACK,SP       ;RESET STACK

032226  012737  000075  004172  MOV     #TTNO,@#TSTNM   ;THIS SAVES TEST NUMBER
```

```
                                                JSR     PC,@#CLDISK     ;CLEAR DISK
                                                MOV     #RHCA,R0
                                        1$:     MOV     (R0)+,R5        ;R5 HAS ADDRESS OF REG. UNDER TEST
                                        2$:     BIS     #CLR,@RHCS2
                                                MOV     @#UNIT,@RHCS2   ;REINSTATE UNIT NO.

                                                ;SET UP FOR AN OPERATION (WRITE HEADER AND DATA)

                                                MOV     @#WRIFOR,@RHCS1 ;WRITE HEADER AND DATA=62
                                                                        ;IN RHCS1
                                                MOV     #-10.,@RHWC     ;10 WORDS
                                                MOV     #WRFROM,@RHBA   ;BUS ADDRESS = WRFROM
                                                MOV     #10,@RHDST      ;DESIRED TRACK=0, SECTOR=10
                                                BIS     #BAI,@RHCS2     ;BUS ADDRESS INCREMENT INHIBIT
                                                MOV     #FMT22,@RHOF    ;FORMAT 16 BIT WORDS
                                                CLR     @RHCA           ;CYLINDER =0

                                                ;SAVE REGISTERS

                                                JSR     R0,@#SAVER      ;SAVE
                                                RHCS1                   ;FROM
                                                REINTO                  ;TO
                                                14.                     ;NUMBER OF REGISTERS SAVED

                                                ;NOW THE COMMAND IS GIVES TO
                                                ;WRITE HEADER AND DATA FOR CYL=0, SECTOR=10
                                                ;TRACK=0 IT COMES BACK AFTER ONE SECTOR
                                                ;HAS PASSED

                                                MOV     #DMD,@RHMR      ;SET DIAGNOSTIC MODE
                                                INC     @RHCS1          ;GO TO RHCS1 WITH 62
                                                MOV     #177672,@R5     ;TRY WRITING ALL BITS EXCEPT
                                                                        ;GO, RMR, IE
                                                BIS     #DMD,@#REINTO+20 ;SET DMD IN SAVED REGISTER RHMR
                                                BIS     #RMR,@#REINTO+2 ;SET RMR IN SAVED REG. RHER1
                                                BIC     #DRY,@#REINTO+22        ;CLEAR DRY IN RHDS1
                                                BIS     #ERR,@#REINTO+22        ;SET ERR IN RHDS1
                                                BIS     #GO,@#REINTO    ;SET GO IN SAVED REG. RHCS1
                                                BIC     #RDY,@#REINTO   ;CLEAR RDY BIT

                                                ;AFTER AN ATTEMPT TO WRITE INTO A REGISTER
                                                ;SAVE REGISTERS AGAIN SO THAT COMPARES CAN BE DONE

                                                JSR     R0,@#SAVER      ;SAVE
                                                RHCS1                   ;FROM
                                                WRFROM                  ;TO
                                                14.                     ;NUMBER

                                                ;AS UPPER BYTE OF RHAS CAN BE CHANGING IN A DUAL PORT
                                                ;OPERATION THE UPPER BYTE OF RHAS WILL BE SAVED AS IS
                                                ;SO THAT THE COMPARES ARE ONLY VALID FOR THE LOWER BYTE
                                                MOVB    @#REINTO+17,@#WRFROM+17;SAVE UPPER RHAS


                                                ;COMPARE REGISTERS BEFORE ATTEMPTED WRITE WITH AFTER
```

```
8390
8391  032446  004037  040736              JSR     R0,@#COMPAR     ;COMPAR
8392  032452  003126                      REINTO                  ;GO BUFFER
8393  032454  002062                      WRFROM                  ;TEST BUFFER
8394  032456  000016                      14.                     ;NUMBER
8395  032460  032466                      4$                      ;RETURN FOR ERROR
8396  032462  032466                      4$                      ;SAME
8397  032464  032506                      5$                      ;RETURN FOR GOOD COMPARISON
8398  032466  013705  044532      4$:     MOV     @#ERWORD,R5     ;GETTING READY TO INDEX
8399  032472  060505                      ADD     R5,R5           ;DOUBLE ERROR WORD
8400  032474  016537  001626  037534      MOV     RHCS1-2(R5),@#REGADR ;FAILING REG. ADDRESS
8401  032502  104001                      ERROR   1               ;CONTENTS OF REGISTER
8402  032504  000207                      RTS     PC              ;CHANGED WITH
8403                                                              ;AN ATTEMPT TO WRITE
8404                                                              ;DURING AN OPERATION
8405                                      ;THE FOLLOWING CLEAR MAY SET THE ATA BIT BECAUSE GO IS HIGH
8406                                      ;
8407  032506  004737  040064      5$:     JSR     PC,@#CLDISK     ;CLEAR DISK
8408
8409
8410
8411
8412                              ;;**************************************************************
8413                              ;*TEST 76       MAKE CURRENT CYLINDER = 1
8414                              ;;**************************************************************
8415  032512  000004              TST76:  SCOPE
8416  032514  012706  001000              MOV     #STACK,SP       ;RESET STACK
8417  032520  004737  040064              JSR PC,@#CLDISK         ;INIT DRIVE
8418  032524  012777  000001  147116      MOV     #DMD,@RHMR              ;SET DIAGNOSTIC MODE
8419  032532  004037  042400              JSR     R0,@#MAKECYL            ;SUBROUTINE TO GIVE A SEEK
8420                                                                      ;COMMAND FOLOWED BY A INIT
8421                                                                      ;THIS SHUOLD CHANGE RHCC
8422  032536  000001                      1                               ;CHANGE RHCC TO 1
8423
8424
8425
8426
8427                              ;;**************************************************************
8428                              ;*TEST 77       ERROR REG1 - BIT #7 - HEADER COMPARE ERROR
8429
8430                              ;*      THE SIMULATED DISK IS SET TO READ CYLINDER=0, TRACK=1
8431                              ;*      SECTOR=1, KEYS=1, 256 WORDS OF 177400
8432                              ;*      A READ HEADER AND DATA COMMAND IS GIVEN TO READ
8433                              ;*      CYLINDER=1, TRACK=1, SECTOR=1, KEY1=1, KEY2=1
8434                              ;*      REINTO BUFFER IS FILLED WITH 0
8435                              ;*      WRFROM IS FILLED WITH 10000,401,1,1,1, AND ALL 177400
8436                              ;*      AFTER THE READ THE REINTO BUFFER IS EXPECTED TO
8437                              ;*      HAVE WHAT IS IN WRFROM - 10000,401,1,1 AND ALL 177400
8438
8439                              ;;**************************************************************
8440  032540  000004              TST77:  SCOPE
8441
8442
8443  032542  012706  001000              MOV     #STACK,SP       ;RESET STACK
8444
8445
```

```
8446  032546  012737  000077  004172          MOV     #TTNO,@#TSTNM   ;THIS SAVES TEST NUMBER
8447
8448
8449  032554  005037  044526                  CLR     @#NOSYNC                ;SET FLAG SO THAT DATA SYNC
8450                                                                          ;AND DATA IS  READ
8451                                          ;FILL SIMULATED DISK
8452  032560  004737  041516                  JSR     PC,@#SETDSK     ;SET UP SIMULATED DISK
8453                                          ;FILL REINTO BUFFER WITH 0
8454
8455  032564  004037  040002                  JSR     R0,@#CLAREA     ;FILL REINTO
8456  032570  003126                          REINTO                  ;FROM
8457  032572  004126                          REINTO+<256.*2> ;TO
8458  032574  000000                          0                       ;DATA
8459
8460                                          ;FILL WRFROM WITH 10000,401,1,1, AND ALL 177400
8461
8462  032576  012700  002062                  MOV     #WRFROM,R0
8463  032602  012720  010000                  MOV     #FMT22,(R0)+    ;10000 INTO WRFROM
8464  032606  012720  000401                  MOV     #401,(R0)+      ;401=TRACK1,SECTOR1
8465  032612  012720  000001                  MOV     #1,(R0)+        ;1 INTO WRFROM+
8466  032616  012720  000001                  MOV     #1,(R0)+        ;1 INTO WRFROM+6
8467
8468                                          ;FILL ALL 0
8469
8470  032622  004037  040002                  JSR     R0,@#CLAREA     ;FILL WRFROM
8471  032626  002072                          WRFROM+10               ;FROM
8472  032630  003062                          WRFROM+<256.*2>         ;TO
8473  032632  177400                          177400                  ;DATA
8474
8475                                          ;NOW GIVE A READ HEADER AND DATA COMMAND
8476                                          ;CYLINDER=1
8477                                          ;TRACK = 1
8478                                          ;SECTOR = 1
8479
8480  032634  004037  041644                  JSR     R0,@#HCCRCE
8481  032640  000072                          72                      ;READ HEADER AND DATA
8482  032642  000001                          1                       ;CYLINDER
8483  032644  000001                          1                       ;SECTOR
8484  032646  000001                          1                       ;TRACK
8485  032650  177400                          -256.                   ;WORD COUNT
8486  032652  003126                          REINTO                  ;RHBA BUFFER
8487  032654  000000                          0                       ;READ
8488
8489  032656  000001                          1                       ;HEADER COMPARE
8490  032660  000240                  1$:     NOP                     ;RETURN POINT FROM HCCRCE
8491
8492
8493
8494
8495
8496                                  ;;*****************************************************************
8497                                  ;*TEST 100      MAKE CURRENT CYLINDER = 0
8498                                  ;;*****************************************************************
8499  032662  000004                  TST100: SCOPE
8500  032664  012706  001000                  MOV     #STACK,SP       ;RESET STACK
8501  032670  004737  040064                  JSR PC,@#CLDISK         ;INIT DRIVE
```

```
8502  032674  012777  000001  146746          MOV     #DMD,@RHMR              ;SET DIAGNOSTIC MODE
8503  032702  004037  042400                  JSR     R0,@#MAKECYL            ;SUBROUTINE TO GIVE A SEEK
8504                                                                          ;COMMAND FOLOWED BY A INIT
8505                                                                          ;THIS SHUOLD CHANGE RHCC
8506  032706  000000                          0                               ;CHANGE RHCC TO 0
8507
8508
8509
8510
8511
8512                                  ;;****************************************************************
8513                                  ;*TEST 101      ERROR REG1 - BIT #7 - HEADER COMPARE ERROR
8514
8515                                  ;*      THE SIMULATED DISK IS SET TO READ CYLINDER=0, TRACK=1
8516                                  ;*      SECTOR=1, KEYS=1, 256 WORDS OF 177400
8517                                  ;*      A READ HEADER AND DATA COMMAND IS GIVEN TO READ
8518                                  ;*      CYLINDER=0, TRACK=0, SECTOR=1, KEY1=1, KEY2=1
8519                                  ;*      REINTO BUFFER IS FILLED WITH 0
8520                                  ;*      WRFROM IS FILLED WITH 10000,401,1,1,1, AND ALL 177400
8521                                  ;*      AFTER THE READ THE REINTO BUFFER IS EXPECTED TO
8522                                  ;*      HAVE WHAT IS IN WRFROM - 10000,401,1,1 AND ALL 177400
8523
8524                                  ;;****************************************************************
8525  032710  000004                  TST101: SCOPE
8526
8527
8528  032712  012706  001000                  MOV     #STACK,SP               ;RESET STACK
8529
8530
8531  032716  012737  000101  004172          MOV     #TTNO,@#TSTNM           ;THIS SAVES TEST NUMBER
8532
8533
8534  032724  005037  044526                  CLR     @#NOSYNC                        ;SET FLAG SO THAT DATA SYNC
8535                                                                                  ;AND DATA IS  READ
8536                                          ;FILL SIMULATED DISK
8537  032730  004737  041516                  JSR     PC,@#SETDSK             ;SET UP SIMULATED DISK
8538                                          ;FILL REINTO BUFFER WITH 0
8539
8540  032734  004037  040002                  JSR     R0,@#CLAREA             ;FILL REINTO
8541  032740  003126                          REINTO                          ;FROM
8542  032742  004126                          REINTO+<256.*2> ;TO
8543  032744  000000                          0                               ;DATA
8544
8545                                          ;FILL WRFROM WITH 10000,401,1,1, AND ALL 177400
8546
8547  032746  012700  002062                  MOV     #WRFROM,R0
8548  032752  012720  010000                  MOV     #FMT22,(R0)+            ;10000 INTO WRFROM
8549  032756  012720  000401                  MOV     #401,(R0)+              ;401=TRACK1,SECTOR1
8550  032762  012720  000001                  MOV     #1,(R0)+                ;1 INTO WRFROM+
8551  032766  012720  000001                  MOV     #1,(R0)+                ;1 INTO WRFROM+6
8552
8553                                          ;FILL ALL 0
8554
8555  032772  004037  040002                  JSR     R0,@#CLAREA             ;FILL WRFROM
8556  032776  002072                          WRFROM+10                       ;FROM
8557  033000  003062                          WRFROM+<256.*2>                 ;TO
```

```
8558  033002  177400                          177400                  ;DATA
8559
8560                                          ;NOW GIVE A READ HEADER AND DATA COMMAND
8561                                          ;CYLINDER=0
8562                                          ;TRACK = 0
8563                                          ;SECTOR = 1
8564
8565  033004  004037  041644                  JSR     R0,@#HCCRCE
8566  033010  000072                          72                      ;READ HEADER AND DATA
8567  033012  000000                          0                       ;CYLINDER
8568  033014  000001                          1                       ;SECTOR
8569  033016  000000                          0                       ;TRACK
8570  033020  177400                          -256.                   ;WORD COUNT
8571  033022  003126                          REINTO                  ;RHBA BUFFER
8572  033024  000000                          0                       ;READ
8573
8574  033026  000001                          1                       ;HEADER COMPARE
8575  033030  000240                  1$:     NOP                     ;RETURN POINT FROM HCCRCE
8576
8577
8578
8579                                  ;;**************************************************************
8580                                  ;*TEST 102      MAKE CURRENT CYLINDER = 1
8581                                  ;;**************************************************************
8582  033032  000004                  TST102: SCOPE
8583  033034  012706  001000                  MOV     #STACK,SP       ;RESET STACK
8584  033040  004737  040064                  JSR PC,@#CLDISK         ;INIT DRIVE
8585  033044  012777  000001  146576          MOV     #DMD,@RHMR              ;SET DIAGNOSTIC MODE
8586  033052  004037  042400                  JSR     R0,@#MAKECYL            ;SUBROUTINE TO GIVE A SEEK
8587                                                                          ;COMMAND FOLOWED BY A INIT
8588                                                                          ;THIS SHUOLD CHANGE RHCC
8589  033056  000001                          1                       ;CHANGE RHCC TO 1
8590
8591
8592
8593
8594
8595
8596
8597                                  ;;**************************************************************
8598                                  ;*TEST 103      ERROR REG.1 - BIT #7 - HEADER COMPARE ERROR
8599
8600                                  ;*      THE SIMULATED DISK IS SET UP FOR CYLINDER=0, TRACK=1
8601                                  ;*      SECTOR=1, KEYS=1, 256 WORDS OF 177400
8602                                  ;*      A WRITE DATA COMMAND IS GIVEN TO WRITE CYLINDER=1
8603                                  ;*      TRACK=1, SECTOR=1, KEY1=1, KEY2=1
8604                                  ;*      WRFROM BUFFER IS FILLED WITH 125252
8605                                  ;*      REINTO BUFFER IS FILLED WITH 177400
8606                                  ;*      AFTER THE WRITE COMMAND THE DISK IS EXPECTED TO
8607                                  ;*      HAVE 177400
8608
8609                                  ;;**************************************************************
8610  033060  000004                  TST103: SCOPE
8611
8612
8613  033062  012706  001000                  MOV     #STACK,SP       ;RESET STACK
```

```
                                ;THIS SAVES TEST NUMBER
                                ;SET FLAG SO THAT DATA SYNC
                                ;AND DATA IS NOT READ
                                ;SETUP SIMULATED DISK
                                ;FILL WRFROM WITH 125252
                                ;FILL WRFROM
                                ;FROM
                                ;TO
                                ;DATA
                                ;FILL REINTO WITH 256 WORDS OF 177400
                                ;THIS IS WHAT IS EXPECTED TO BE ON DISK EVEN AFTER
                                ;AN ATTEMPT TO WRITE 125252
                                ;FILL REINTO
                                ;FROM
                                ;TO
                                ;NOW GI Z A WRITE DATA COMMAND
                                ;CYLINDER = 1,
                                ;TRACK = 1
                                ;SECTOR = 1
                                ;WRITE DATA
                                ;CYLINDER
                                ;SECTOR
                                ;TRACK
                                ;WORD COUNT
                                ;RHBA BUFFER
                                ;WRITE
                                ;HEADER COMPARE
                                ;RETURN POINT FROM HCCRCE
                                ;;*****************************************************************
                                ;*TEST 104      MAKE CURRENT CYLINDER = 0
                                ;;*****************************************************************
                                ;RESET STACK
                                ;INIT DRIVE
                                ;SET DIAGNOSTIC MODE
```

```
8670  033200  004037  042400                  JSR     R0,@#MAKECYL            ;SUBROUTINE TO GIVE A SEEK
8671                                                                          ;COMMAND FOLOWED BY A INIT
8672                                                                          ;THIS SHUOLD CHANGE RHCC
8673  033204  000000                          0                               ;CHANGE RHCC TO 0
8674
8675
8676
8677
8678                                    ;;**********************************************************************
8679                                    ;*TEST 105      ERROR REG.1 - BIT #7 - HEADER COMPARE ERROR
8680
8681                                    ;*      THE SIMULATED DISK IS SET UP FOR CYLINDER=0, TRACK=1
8682                                    ;*      SECTOR=1, KEYS=1, 256 WORDS OF 177400
8683                                    ;*      A WRITE DATA COMMAND IS GIVEN TO WRITE CYLINDER=0
8684                                    ;*      TRACK=0, SECTOR=1, KEY1=1, KEY2=1
8685                                    ;*      WRFROM BUFFER IS FILLED WITH 125252
8686                                    ;*      REINTO BUFFER IS FILLED WITH 177400
8687                                    ;*      AFTER THE WRITE COMMAND THE DISK IS EXPECTED TO
8688                                    ;*      HAVE 177400
8689
8690                                    ;;**********************************************************************
8691  033206  000004                    TST105: SCOPE
8692
8693
8694  033210  012706  001000                    MOV     #STACK,SP       ;RESET STACK
8695
8696
8697  033214  012737  000105  004172            MOV     #TTNO,@#TSTNM   ;THIS SAVES TEST NUMBER
8698
8699
8700  033222  012737  177777  044526      MOV     #-1,@#NOSYNC                ;SET FLAG SO THAT DATA SYNC
8701                                                                          ;AND DATA IS NOT READ
8702                                      ;FILL SIMULATED DISK
8703
8704  033230  004737  041516                    JSR     PC,@#SETDSK     ;SETUP SIMULATED DISK
8705
8706
8707                                      ;FILL WRFROM WITH 125252
8708
8709  033234  004037  040002                    JSR     R0,@#CLAREA     ;FILL WRFROM
8710  033240  002062                            WRFROM                  ;FROM
8711  033242  003062                            WRFROM+<256.*2>         ;TO
8712  033244  125252                            125252                  ;DATA
8713
8714                                      ;FILL REINTO WITH 256 WORDS OF 177400
8715                                      ;THIS IS WHAT IS EXPECTED TO BE ON DISK EVEN AFTER
8716                                      ;AN ATTEMPT TO WRITE 125252
8717
8718  033246  004037  040002                    JSR     R0,@#CLAREA     ;FILL REINTO
8719  033252  003126                            REINTO                  ;FROM
8720  033254  004126                            REINTO+<256.*2>         ;TO
8721  033256  177400                            177400
8722
8723                                      ;NOW GIVE A WRITE DATA COMMAND
8724                                      ;CYLINDER = 0,
8725                                      ;TRACK = 0
```

```
                                        ;SECTOR = 1
                                        JSR     R0,@#HCCRCE
033260  004037  041644                  60                      ;WRITE DATA
033264  000060                          0                       ;CYLINDER
033266  000000                          1                       ;SECTOR
033270  000001                          0                       ;TRACK
033272  000000                          -256.                   ;WORD COUNT
033274  177400                          WRFROM                  ;RHBA BUFFER
033276  002062                          1                       ;WRITE
033300  000001
                                        1                       ;HEADER COMPARE
033302  000001                  1$:     NOP                     ;RETURN POINT FROM HCCRCE
033304  000240

                                ;;*****************************************************************
                                ;*TEST 106       ERROR REG.1 - BIT #8 - CRC ERROR

                                ;*      THE SIMULATED DISK IS SET TO READ CYLINDER=0, TRACK=1
                                ;*      SECTOR=1, KEYS=1, 256 WORDS OF 177400
                                ;*      A READ HEADER AND DATA COMMAND IS GIVEN TO READ
                                ;*      CYLINDER=0, TRACK=1, SECTOR=1, KEY1=1, KEY2=1
                                ;*      REINTO BUFFER IS FILLED WITH 0
                                ;*      WRFROM IS FILLED WITH 10000,401,1,1,1, AND ALL 177400
                                ;*      AFTER THE READ THE REINTO BUFFER IS EXPECTED TO
                                ;*      HAVE WHAT IS IN WRFROM - 10000,401,1,1 AND ALL 177400

                                ;;*****************************************************************
033306  000004                  TST106: SCOPE


033310  012706  001000                  MOV     #STACK,SP       ;RESET STACK

033314  012737  000106  004172          MOV     #TTNO,@#TSTNM   ;THIS SAVES TEST NUMBER

033322  005037  044526                  CLR     @#NOSYNC                ;SET FLAG SO THAT DATA SYNC
                                                                        ;AND DATA IS  READ
                                        ;FILL SIMULATED DISK
033326  004737  041516                  JSR     PC,@#SETDSK     ;SET UP SIMULATED DISK
033332  005137  046312                  COM     @#WCRC          ;CHANCE CRC TO GIVE HCRC

                                        ;FILL REINTO BUFFER WITH 0

033336  004037  040002                  JSR     R0,@#CLAREA     ;FILL REINTO
033342  003126                          REINTO                  ;FROM
033344  004126                          REINTO+<256.*2> ;TO
033346  000000                          0                       ;DATA

                                        ;FILL WRFROM WITH 10000,401,1,1, AND ALL 177400

033350  012700  002062                  MOV     #WRFROM,R0
033354  012720  010000                  MOV     #FMT22,(R0)+    ;10000 INTO WRFROM
033360  012720  000401                  MOV     #401,(R0)+      ;401=TRACK1,SECTOR1
033364  012720  000001                  MOV     #1,(R0)+        ;1 INTO WRFROM+
```

```
8782  033370  012720  000001                  MOV     #1,(R0)+        ;1 INTO WRFROM+6
8783
8784                                          ;FILL ALL 0
8785
8786  033374  004037  040002                  JSR     R0,@#CLAREA     ;FILL WRFROM
8787  033400  002072                          WRFROM+10               ;FROM
8788  033402  003062                          WRFROM+<256.*2>         ;TO
8789  033404  177400                          177400                  ;DATA
8790
8791                                          ;NOW GIVE A READ HEADER AND DATA COMMAND
8792                                          ;CYLINDER=0
8793                                          ;TRACK = 1
8794                                          ;SECTOR = 1
8795
8796  033406  004037  041644                  JSR     R0,@#HCCRCE
8797  033412  000072                          72                      ;READ HEADER AND DATA
8798  033414  000000                          0                       ;CYLINDER
8799  033416  000001                          1                       ;SECTOR
8800  033420  000001                          1                       ;TRACK
8801  033422  177400                          -256.                   ;WORD COUNT
8802  033424  003126                          REINTO                  ;RHBA BUFFER
8803  033426  000000                          0                       ;READ
8804
8805  033430  000000                          0                       ;CRC ERROR
8806  033432  000240                  1$:     NOP                     ;RETURN POINT FROM HCCRCE
8807
8808
8809                                          ;;*****************************************************************
8810                                          ;*TEST 107      ERROR REG.1 - BIT #8 - CRC ERROR
8811
8812                                          ;*      THE SIMULATED DISK IS SET TO READ CYLINDER=0, TRACK=1
8813                                          ;*      SECTOR=1, KEYS=1, 256 WORDS OF 177400
8814                                          ;*      A READ HEADER AND DATA COMMAND IS GIVEN TO READ
8815                                          ;*      CYLINDER=0, TRACK=1, SECTOR=1, KEY1=1, KEY2=1
8816                                          ;*      REINTO BUFFER IS FILLED WITH 0
8817                                          ;*      WRFROM IS FILLED WITH 10000,401,1,1,1, AND ALL 177400
8818                                          ;*      AFTER THE READ THE REINTO BUFFER IS EXPECTED TO
8819                                          ;*      HAVE WHAT IS IN WRFROM - 10000,401,1,1 AND ALL 177400
8820
8821                                          ;;*****************************************************************
8822  033434  000004                  TST107: SCOPE
8823
8824
8825  033436  012706  001000                  MOV     #STACK,SP       ;RESET STACK
8826
8827
8828  033442  012737  000107  004172          MOV     #TTNO,@#TSTNM   ;THIS SAVES TEST NUMBER
8829
8830
8831  033450  005037  044526                  CLR     @#NOSYNC                ;SET FLAG SO THAT DATA SYNC
8832                                                                          ;AND DATA IS  READ
8833                                          ;FILL SIMULATED DISK
8834  033454  004737  041516                  JSR     PC,@#SETDSK     ;SET UP SIMULATED DISK
8835  033460  005137  046312                  COM     @#WCRC          ;CHANCE CRC TO GIVE HCRC
8836
8837                                          ;FILL REINTO BUFFER WITH 0
```

```
                                ;FILL REINTO WITH 10000,401,1,1, AND ALL 177400

033476  012700  002062          MOV     #WRFROM,R0
033502  012720  010000          MOV     #FMT22,(R0)+    ;10000 INTO WRFROM
033506  012720  000401          MOV     #401,(R0)+      ;401=TRACK1,SECTOR1
033512  012720  000001          MOV     #1,(R0)+        ;1 INTO WRFROM+
033516  012720  000001          MOV     #1,(R0)+        ;1 INTO WRFROM+6

                                ;FILL ALL 0

033522  004037  040002          JSR     R0,@#CLAREA     ;FILL WRFROM
033526  002072                  WRFROM+10               ;FROM
033530  003062                  WRFROM+<256.*2>         ;TO
033532  177400                  177400                  ;DATA

                                ;NOW GIVE A READ HEADER AND DATA COMMAND
                                ;CYLINDER=0
                                ;TRACK = 1
                                ;SECTOR = 1

033534  004037  041644          JSR     R0,@#HCCRCE
033540  000072                  72                      ;READ HEADER AND DATA
033542  000000                  0                       ;CYLINDER
033544  000001                  1                       ;SECTOR
033546  000001                  1                       ;TRACK
033550  177400                  -256.                   ;WORD COUNT
033552  003126                  REINTO                  ;RHBA BUFFER
033554  000000                  0                       ;READ

033556  000000                  0                       ;CRC ERROR
033560  000240          1$:     NOP                     ;RETURN POINT FROM HCCRCE

                                ;;*****************************************************************
                                ;*TEST 110      ERROR REG.1 - BIT 8 - CRC ERROR

                                ;*      THE SIMULATED DISK IS SET UP FOR CYLINDER=0, TRACK=1
                                ;*      SECTOR=1, KEYS=1, 256 WORDS OF 177400
                                ;*      A WRITE DATA COMMAND IS GIVEN TO WRITE CYLINDER=0
                                ;*      TRACK=1, SECTOR=1, KEY1=1, KEY2=1
                                ;*      WRFROM BUFFER IS FILLED WITH 125252
                                ;*      REINTO BUFFER IS FILLED WITH 177400
                                ;*      AFTER THE WRITE COMMAND THE DISK IS EXPECTED TO
                                ;*      HAVE 177400

                                ;;*****************************************************************
033562  000004          TST110: SCOPE

033564  012706  001000          MOV     #STACK,SP       ;RESET STACK
```

```
8894
8895
8896  033570  012737  000110  004172          MOV     #TTNO,@#TSTNM   ;THIS SAVES TEST NUMBER
8897
8898
8899  033576  012737  177777  044526          MOV     #-1,@#NOSYNC            ;SET FLAG SO THAT DATA SYNC
8900                                                                          ;AND DATA IS NOT READ
8901                                          ;FILL SIMULATED DISK
8902
8903  033604  004737  041516                  JSR     PC,@#SETDSK     ;SETUP SIMULATED DISK
8904
8905  033610  005137  046312                  COM     @#WCRC          ;CHANGE CRC TO GIVE HCRC
8906
8907
8908                                          ;FILL WRFROM WITH 125252
8909
8910  033614  004037  040002                  JSR     R0,@#CLAREA     ;FILL WRFROM
8911  033620  002062                          WRFROM                  ;FROM
8912  033622  003062                          WRFRO.+<256.*2>         ;TO
8913  033624  125252                          125252                  ;DATA
8914
8915                                          ;FILL REINTO WITH 256 WORDS OF 177400
8916                                          ;THIS IS WHAT IS EXPECTED TO BE ON DISK EVEN AFTER
8917                                          ;AN ATTEMPT TO WRITE 125252
8918
8919  033626  004037  040002                  JSR     R0,@#CLAREA     ;FILL REINTO
8920  033632  003126                          REINTO                  ;FROM
8921  033634  004126                          REINTO+<256.*2>         ;TO
8922  033636  177400                          177400
8923
8924                                          ;NOW GIVE A WRITE DATA COMMAND
8925                                          ;CYLINDER = 0,
8926                                          ;TRACK = 1
8927                                          ;SECTOR = 1
8928
8929  033640  004037  041644                  JSR     R0,@#HCCRCE
8930  033644  000060                          60                      ;WRITE DATA
8931  033646  000000                          0                       ;CYLINDER
8932  033650  000001                          1                       ;SECTOR
8933  033652  000001                          1                       ;TRACK
8934  033654  177400                          -256.                   ;WORD COUNT
8935  033656  002062                          WRFROM                  ;RHBA BUFFER
8936  033660  000001                          1                       ;WRITE
8937
8938  033662  000000                          0                       ;CRC ERROR
8939  033664  000240          1$:             NOP                     ;RETURN POINT FROM HCCRCE
8940
8941
8942                                          ;;*****************************************************************
8943                                          ;*TEST 111       ERROR REG.1 - BIT #8 - CRC ERROR
8944
8945                                          ;*      THE SIMULATED DISK IS SET UP FOR CYLINDER=0, TRACK=1
8946                                          ;*      SECTOR=1, KEYS=1, 256 WORDS OF 177400
8947                                          ;*      A WRITE DATA COMMAND IS GIVEN TO WRITE CYLINDER=0
8948                                          ;*      TRACK=1, SECTOR=1, KEY1=1, KEY2=1
8949                                          ;*      WRFROM BUFFER IS FILLED WITH 125252
```

```
                                        ;*      REINTO BUFFER IS FILLED WITH 177400
                                        ;*      AFTER THE WRITE COMMAND THE DISK IS EXPECTED TO
                                        ;*      HAVE 177400
                                        ;
                                        ;;*****************************************************************
033666  000004                          TST111: SCOPE

033670  012706  001000                          MOV     #STACK,SP       ;RESET STACK

033674  012737  000111  004172                  MOV     #TTNO,@#TSTNM   ;THIS SAVES TEST NUMBER

033702  012737  177777  044526                  MOV     #-1,@#NOSYNC            ;SET FLAG SO THAT DATA SYNC
                                                                                ;AND DATA IS NOT READ
                                        ;FILL SIMULATED DISK

033710  004737  041516                          JSR     PC,@#SETDSK     ;SETUP SIMULATED DISK

033714  005137  046312                          COM     @#WCRC          ;CHANGE CRC TO GIVE HCRC


                                        ;FILL WRFROM WITH 125252

033720  004037  040002                          JSR     R0,@#CLAREA     ;FILL WRFROM
033724  002062                          WRFROM                          ;FROM
033726  003062                          WRFROM+<256.*2>                 ;TO
033730  125252                          125252                          ;DATA

                                        ;FILL REINTO WITH 256 WORDS OF 177400
                                        ;THIS IS WHAT IS EXPECTED TO BE ON DISK EVEN AFTER
                                        ;AN ATTEMPT TO WRITE 125252

033732  004037  040002                          JSR     R0,@#CLAREA     ;FILL REINTO
033736  003126                          REINTO                          ;FROM
033740  004126                          REINTO+<256.*2>                 ;TO
033742  177400                          177400

                                        ;NOW GIVE A WRITE DATA COMMAND
                                        ;CYLINDER = 0,
                                        ;TRACK = 1
                                        ;SECTOR = 1

033744  004037  041644                          JSR     R0,@#HCCRCE
033750  000060                          60                              ;WRITE DATA
033752  000000                          0                               ;CYLINDER
033754  000001                          1                               ;SECTOR
033756  000001                          1                               ;TRACK
033760  177400                          -256.                           ;WORD COUNT
033762  002062                          WRFROM                          ;RHBA BUFFER
033764  000001                          1                               ;WRITE

033766  000000                          0                               ;CRC ERROR
033770  000240                  1$:     NOP                             ;RETURN POINT FROM HCCRCE
```

```
                                        ;;*****************************************************************
                                        ;*TEST 112      MAKE CURRENT CYLINDER = 410.
                                        ;;*****************************************************************
033772  000004                          TST112: SCOPE
033774  012706  001000                          MOV     #STACK,SP       ;RESET STACK
034000  004737  040064                          JSR PC,@#CLDISK         ;INIT DRIVE
034004  012777  000001  145636                  MOV     #DMD,@RHMR              ;SET DIAGNOSTIC MODE
034012  004037  042400                          JSR     R0,@#MAKECYL            ;SUBROUTINE TO GIVE A SEEK
                                                                                ;COMMAND FOLOWED BY A INIT
                                                                                ;THIS SHUOLD CHANGE RHCC
034016  000632                                  410.                            ;CHANGE RHCC TO 410.


                                        ;;*****************************************************************
                                        ;*TEST 113      RHDS1 - BIT #10 LAST SECTOR TRANSFERRED
                                        ;*      WRITE CYLINDER 410., FORMAT 16 BITS PER WORD
                                        ;*      TRACK 18., SECTOR 21., KEYS 0, NUMBER OF WORDS
                                        ;*      256., OF 377
                                        ;*      LST BIT # 10 RHDS1 SHOULD SET AFTER WRITE
                                        ;*      IS COMPLETE.
                                        ;;*****************************************************************
034020  000004                          TST113: SCOPE

034022  012706  001000                          MOV     #STACK,SP       ;RESET STACK

034026  012737  000113  004172                  MOV     #TTNO,@#TSTNM   ;THIS SAVES TEST NUMBER

034034  004037  040002                          JSR     R0,@#CLAREA     ;CLEAR SIMULATED DISK
034040  046330                                  .WORD   DISK            ;FROM
034042  047354                                  .WORD   TOLGAP+16       ;TO
034044  000000                                  .WORD   0               ;DATA
                                        ;THESE ARE SETUP FOR DISKLESS USE ONLY
034046  012737  010632  044412                  MOV     #410.!FMT22,@#CYL;CYLINDER 410.
                                                                        ;16 BITS PER WORD
034054  112737  000022  044415                  MOVB    #18.,@#SECOTR+1 ;TRACK 18.
034062  112737  000025  044414                  MOVB    #21.,@#SECOTR   ;SECTOR 21.
034070  005037  044416                          CLR     @#KEY1          ;KEY1 0
034074  005037  044420                          CLR     @#KEY2          ;KEY2 0
034100  012737  000400  044460                  MOV     #256.,@#NOWORD  ;NO OF DATA WORDS
034106  012737  000001  044422                  MOV     #1,@#X          ;WRITE DATA
034114  004537  041242                          JSR     R5,@#CRC        ;GO TO CALCULATE CRC
034120  044412                                  CYL
034122  046312                                  WCRC


                                        ;THESE ARE REGULAR SETUPS
```

```
 9062
 9063
 9064  034124  004037  040002                    JSR     R0,@#CLAREA     ;FILL WRITE BUFFER WITH 377
 9065  034130  002062                            WRFROM                  ;FROM
 9066  034132  003062                            WRFROM+<256.*2>         ;TO
 9067  034134  000377                            377                     ;DATA
 9068  034136  004737  040064                    JSR     PC,@#CLDISK     ;SETUP GENERAL REGISTERS
 9069  034142  012777  177400  145452            MOV     #-256.,@RHWC    ;256. DATA WORDS
 9070  034150  012777  002062  145446            MOV     #WRFROM,@RHBA   ;STARTING ADDRESS OF WRITE BUFFER
 9071  034156  012746  000025                    MOV     #21.,-(SP)      ;SECTOR 21.
 9072  034162  112766  000022  000001            MOVB    #18.,1(SP)      ;TRACK 18.
 9073  034170  012677  145440                    MOV     (SP)+,@RHDST    ;SECTOR 21. TRACK 18.
 9074  034174  012777  010000  145436            MOV     #FMT22,@RHOF    ;16 BITS PER WORD FORMAT
 9075  034202  012777  000632  145432            MOV     #410.,@RHCA     ;CYLINDER 410.
 9076  034210  004737  040120                    JSR     PC,@#CHECKT     ;CHECK FOR DVA,RDY,DPR,DRY
 9077  034214  013711  002036                    MOV     @#WRIDAT,@R1    ;WRITE DATA=60
 9078  034220  005037  001772                    CLR     @#ERFLG$        ;CLEAR ERROR FLAG
 9079  034224  004737  044302                    JSR     PC,@#COMHD      ;WRITE DATA
 9080                                    ;IF THE PROGRAM COMES BACK HERE WITHOUT ERROR PRINTOUTS
 9081                                    ;FROM THE "COMHD" ROUTINE IT MEANS SECTOR GAP, SYNC BYTE
 9082                                    ;HEADER, HEADER CRC, HEADER GAP AND SYNC BYTE HAVE GONE BY
 9083                                    ;AND SYNCS WERE CORRECTLY DETECTED
 9084                                    ;DATA IS TO BE CHECKED
 9085  034230  004737  037470                    JSR     PC,@#PUTREG     ;SAVE REGISTERS
 9086  034234  005737  001772                    TST     @#ERFLG$                ;HAS ANY ERRORS OCCURED?
 9087  034240  001062                            BNE     5$      ;BRANCH IF YES
 9088  034242  012700  000377                    MOV     #377,R0         ;GOOD DATA
 9089  034246  012701  046330                    MOV     #DISK,R1        ;DATA WRITTEN INTO "DISK"
 9090  034252  012702  000400                    MOV     #256.,R2                ;COUNTER
 9091  034256  012737  000401  044532  1$:       MOV     #256.+1,@#ERWORD        ;FOR ERROR WORD
 9092  034264  020021                            CMP     R0,(R1)+        ;COMPARE GOOD DATA WITH DATA ON DISK
 9093  034266  001424                            BEQ     3$              ;BRANCH IF GOOD
 9094  034270  010037  001124                    MOV     R0,@#$GDDAT     ;GOOD DATA
 9095  034274  014137  001126                    MOV     -(R1),@#$BDDAT  ;BAD DATA
 9096  034300  160237  044532                    SUB     R2,@#ERWORD     ;ERROR WORD NO
 9097  034304  005737  001772                    TST     @#ERFLG$                ;ANY ERRORS ALREADY THERE?
 9098  034310  001002                            BNE     2$              ;BRANCH IF YES
 9099  034312  104004                            ERROR   4               ;ERROR ON WRITE DATA COMMAND
 9100  034314  000401                            BR      64$             ;BRANCH TO AVOID PRINTING NEXT ERROR
 9101  034316  104005                    2$:     ERROR   5               ;WORD NO GIVES WORD IN ERROR
 9102  034320  005721                    64$:    TST     (R1)+           ;UNDO -(R1) FOR BAD DATA
 9103  034322  013746  001140                    MOV     @#SWR,-(SP)     ;GET SWITCH SETTING
 9104  034326  042716  177177                    BIC     #177177,(SP)    ;KEEP ONLY SWITCH 7 AND 8
 9105  034332  022726  000200                    CMP     #SW07,(SP)+     ;IS 7 SET AND 8 RESET
 9106  034336  001402                            BEQ     4$              ;BRANCH OUT IF YES
 9107  034340  005302                    3$:     DEC     R2              ;IF NOT COUNT 256 WORDS
 9108  034342  001345                            BNE     1$              ;BRANCH IF 256. NOT DONE
 9109  034344  013746  001722            4$:     MOV     @#DS1,-(SP)     ;GET RHDS1
 9110  034350  042716  001000                    BIC     #PROG,(SP)      ;CLEAR PROG
 9111  034354  022726  002700                    CMP     #LST!DPR!DRY!VV,(SP)+;IS LST HIGH ?
 9112  034360  001412                            BEQ     5$              ;BRANCH IF GOOD
 9113  034362  013737  001652  037534            MOV     @#RHDS1,@#REGADR ;FAILING REG. ADDRESS
 9114  034370  012737  002700  001124            MOV     #LST!DPR!DRY!VV,@#$GDDAT ;GOOD DATA
 9115  034376  013737  001722  001126            MOV     @#DS1,@#$BDDAT  ;BAD DATA
 9116  034404  104001                            ERROR   1               ;LST DID NOT SET AFTER
 9117                                            ;LAST SECTOR ON LAST TRACK
```

```
                                                        ;ON LAST CYLINDER WAS
                                                        ;WRITTEN
                                                        ;VV BIT #6 MAY OR MAY NOT BE HIGH
034406  013737  001630  034416  5$:     MOV     @#RHCS1,@#6$
034414  104412                        WAT
034416  000000                6$:     0               ;RHCS1 ADDRESS
034420  000200                        RDY             ;WAIT FOR READY


                        ;;*****************************************************************
                        ;*TEST 114      ERROR REGISTER 1 - BIT #9 AOE

                        ;*      A WRITE DATA COMMAND IS GIVEN TO CYLINDER 410
                        ;*      SECTOR 21 TRACK 18, KEYS 0, DATA 377
                        ;*      WORD COUNT REGISTER FOR 326 (256+66+4) WORDS
                        ;*
                        ;*      AFTER 256 WORDS HAVE BEEN WRITTEN
                        ;*      AOE SHOULD COME UP
                        ;*      RHWC WILL SHOW 4 BECAUSE THE SILO IS 66 WORDS AND
                        ;*      256 WORDS HAVE BEEN WRITTEN - TOTAL 322
                        ;*      THIS IS 4 SHORT OF 326

                        ;;*****************************************************************
034422  000004         TST114: SCOPE
034424  012706  001000            MOV     #STACK,SP       ;RESET STACK

034430  012737  000114  004172    MOV     #TTNO,@#TSTNM   ;THIS SAVES TEST NUMBER

034436  004737  040064            JSR     PC,@#CLDISK     ;INIT AND SET GENERAL REGISTERS
034442  004037  040002            JSR     R0,@#CLAREA     ;CLEAR SIMULATED DISK
034446  046330                    .WORD   DISK            ;FROM
034450  047354                    .WORD   TOLGAP+16       ;TO
034452  000000                    .WORD   0                       ;DATA
                        ;THESE ARE SETUP FOR DISKLESS USE ONLY
034454  012737  010632  044412    MOV     #410.!FMT22,@#CYL ;CYLINDER 410.
                                                        ;16 BITS PER WORD
034462  112737  000022  044415    MOVB    #18.,@#SECOTR+1 ;TRACK 18.
034470  112737  000025  044414    MOVB    #21.,@#SECOTR   ;SECTOR 21.
034476  005037  044416            CLR     @#KEY1          ;KEY1 0
034502  005037  044420            CLR     @#KEY2          ;KEY2 0
034506  012737  000400  044460    MOV     #256.,@#NOWORD  ;NO OF DATA WORDS
034514  012737  000001  044422    MOV     #1,@#X          ;WRITE DATA
034522  004537  041242            JSR     R5,@#CRC        ;GO TO CALCULATE CRC
034526  044412                    CYL
034530  046312                    WCRC

                        ;THESE ARE REGULAR SETUPS

034532  004037  040002            JSR     R0,@#CLAREA     ;FILL WRITE BUFFER WITH 377
034536  002062                    WRFROM                  ;FROM
034540  003062                    WRFROM+<256.*2>         ;TO
034542  000377                    377                     ;DATA
034544  004737  040064            JSR     PC,@#CLDISK     ;SETUP GENERAL REGISTERS
034550  012777  177272  145044    MOV     #-326.,@RHWC    ;326. DATA WORDS
```

```
9174  034556  012777  002062  145040          MOV     #WRFROM,@RHBA   ;STARTING ADDRESS OF WRITE BUFFER
9175  034564  012746  000025                  MOV     #21.,-(SP)      ;SECTOR 21.
9176  034570  112766  000022  000001          MOVB    #18.,1(SP)      ;TRACK 18.
9177  034576  012677  145032                  MOV     (SP)+,@RHDST    ;SECTOR 21. TRACK 18.
9178  034602  012777  010000  145030          MOV     #FMT22,@RHOF    ;16 BITS PER WORD FORMAT
9179  034610  012777  000632  145024          MOV     #410.,@RHCA     ;CYLINDER 410.
9180  034616  004737  040120                  JSR     PC,@#CHECKT     ;CHECK FOR DVA,RDY,DPR,DRY
9181  034622  013711  002036                  MOV     @#WRIDAT,@R1    ;WRITE DATA=60
9182  034626  005037  001772                  CLR     @#ERFLG$        ;CLEAR ERROR FLAG
9183
9184                                          ;THE REGISTERS WILL BE SAVED IN REINTO BUFFER
9185  034632  004037  040542                  JSR     R0,@#SAVER      ;SAVE
9186  034636  001622                          RHWC                    ;FROM
9187  034640  003126                          REINTO                  ;TO
9188  034642  000023                          19.                     ;NUMBER SAVED
9189
9190                                          ;GIVE WRITE DATA COMMAND
9191  034644  004737  044302                  JSR     PC,@#COMHD      ;WRITE DATA COMMAND
9192
9193                                          ;CHANGE SAVED REGISTERS TO EXPECTED VALUE
9194  034650  012737  177702  003126          MOV     #-76,@#REINTO   ;SAVED RHWC SHOULD BE=76
9195  034656  012737  003102  003130          MOV     #WRFROM+<2*256.>+<2*8.>,@#REINTO+2
9196                                                                  ;SAVED RHBA SHOULD BE WRFROM+256+66
9197  034664  052737  000200  003132          BIS     #OR,@#REINTO+4  ;SAVED RHCS2
9198  034672  042737  000100  003132          BIC     #IR,@#REINTO+4  ;SAVED RHCS2
9199  034700  052737  140000  003134          BIS     #SC!TRE,@#REINTO+6      ;SAVED RHCS1 SHOULD HAVE SC TRE
9200  034706  012737  001000  003136          MOV     #AOE,@#REINTO+10 ;SAVED RHER1 SHOULD HAVE AOE
9201  034714  017737  144714  003140          MOV     @RHDST,@#REINTO+12      ;SAVED RHDST SHOULD HAVE=
9202                                                                  ;RHDST IS UNDEFINED
9203  034722  012737  000633  003146          MOV     #411.,@#REINTO+20       ;SAVED DESIRED CYLINDER ADDRESS
9204  034730  013737  002002  003152          MOV     @#ATTENT,@#REINTO+24 ;SAVED RHAS SHOULD HAVE APPROPIATE BIT
9205  034736  052737  000001  003154          BIS     #DMD,@#REINTO+26        ;SAVED RHMR
9206  034744  052737  142000  003156          BIS     #ATA!ERR!LST,@#REINTO+30 ;SAVED RHDS1
9207
9208                                          ;AFTER A WRITE DATA COMMAND WITH AOE ERROR
9209                                          ;SAVE REGISTERS AGAIN SO THAT COMPARES CAN BE DONE
9210  034752  004037  040542                  JSR     R0,@#SAVER      ;SAVE
9211  034756  001622                          RHWC                    ;FROM
9212  034760  002062                          WRFROM                  ;TO
9213  034762  000021                          17.                     ;NUMBER OF REGISTERS SAVED
9214
9215                                          ;AS UPPER BYTE OF RHAS CAN BE CHANGING IN A DUAL PORT
9216                                          ;OPERATION THE UPPER BYTE OF RHAS WILL BE SAVED AS IS
9217                                          ;SO THAT THE COMPARES ARE ONLY VALID FOR THE LOWER BYTE
9218  034764  113737  003153  002107          MOVB    @#REINTO+25,@#WRFROM+25;SAVE UPPER RHAS
9219
9220
9221                                          ;COMPARE REGISTERS BEFORE WRITE DATA COMMAND
9222                                          ;WITH AFTER COMMAND
9223  034772  004037  040736                  JSR     R0,@#COMPAR     ;COMPARE
9224  034776  003126                          REINTO                  ;GOOD BUFFER
9225  035000  002062                          WRFROM                  ;TEST BUFFER
9226  035002  000021                          17.                     ;NUMBER OF REGISTERS
9227  035004  035012                          1$                      ;RETURN FOR ERROR
9228  035006  035012                          1$                      ;SAME
9229  035010  035032                          2$                      ;RETURN FOR GOOD COMPARISON
```

```
                                1$:     MOV     @#ERWORD,R5     ;GETTING READY TO INDEX
                                        ADD     R5,R5           ;DOUBLE ERROR WORD
                                        MOV     RHWC-2(R5),@#REGADR ;FAILING REG. ADDRESS
                                        ERROR   1               ;FORCED AOE ERROR CAUSED IMPROPER
                                                                ;REGISTER CHANGE
                                        RTS     PC              ;RETURN FOR FURTHER COMPARISONS
                                                                ;NO ERRORS

                                2$:     CLR     @#ERFLG$        ;CLEAR ERROR FLAG

                                        ;DATA IS TO BE CHECKED
                                        JSR     PC,@#PUTREG     ;SAVE REGISTERS
                                        MOV     #377,R0         ;GOOD DATA
                                        MOV     #DISK,R1        ;DATA WRITTEN INTO "DISK"
                                        MOV     #256.,R2        ;COUNTER
                                3$:     MOV     #256.,@#ERWORD  ;FOR ERROR WORD
                                        CMP     R0,(R1)+        ;COMPARE GOOD DATA WITH DATA ON DISK
                                        BEQ     6$              ;BRANCH IF GOOD
                                        MOV     R0,@#$GDDAT     ;GOOD DATA
                                        MOV     -(R1),@#$BDDAT  ;BAD DATA
                                        SUB     R2,@#ERWORD     ;ERROR WORD NO
                                        TST     @#ERFLG$        ;ANY ERRORS ALREADY THERE?
                                        BNE     4$              ;BRANCH IF YES
                                        ERROR   4               ;ERROR ON WRITE DATA COMMAND WITH FORCED AOE
                                        BR      5$              ;BRANCH TO AVOID PRINTING NEXT ERROR
                                4$:     ERROR   5               ;WORD NO GIVES WORD IN ERROR
                                5$:     TST     (R1)+           ;UNDO -(R1) FOR BAD DATA
                                        MOV     @#SWR,-(SP)     ;GET SWITCH SETTING
                                        BIC     #177177,(SP)    ;KEEP ONLY SWITCH 7 AND 8
                                        CMP     #SW07,(SP)+     ;IS 7 SET AND 8 RESET

                                        BEQ     7$              ;BRANCH OUT IF YES
                                6$:     DEC     R2              ;IF NOT COUNT 256 WORDS
                                        BNE     3$              ;BRANCH IF 256. NOT DONE

                                7$:


                                ;;*****************************************************************
                                ;*TEST 115      MAKE CURRENT CYLINDER = 0
                                ;;*****************************************************************
                                TST115: SCOPE
                                        MOV     #STACK,SP       ;RESET STACK
                                        JSR PC,@#CLDISK         ;INIT DRIVE
                                        MOV     #DMD,@RHMR              ;SET DIAGNOSTIC MODE
                                        JSR     R0,@#MAKECYL            ;SUBROUTINE TO GIVE A SEEK
                                                                        ;COMMAND FOLOWED BY A INIT
                                                                        ;THIS SHUOLD CHANGE RHCC
                                        0                               ;CHANGE RHCC TO 0
```

```
                                ;;*****************************************************************
                                ;*TEST 116      ERROR REGISTER 1- BIT #10 IAE

                                ;*      A READ HEADER AND DATA IS GIVEN TO TRACK 20
                                ;*      SECTOR 0
                                ;*
                                ;*      AN INDEX PULSE IS GIVEN TO GET RHLA TO 0
                                ;*
                                ;*      IAE BIT SHOULD SET

                                ;;*****************************************************************
035172  000004                  TST116: SCOPE
035174  012706  001000                  MOV     #STACK,SP       ;RESET STACK

035200  012737  000116  004172          MOV     #TTNO,@#TSTNM   ;THIS SAVES TEST NUMBER

035206  004737  040064                  JSR     PC,@#CLDISK     ;CLEAR REGISTERS AND SET UNIT NO.

                                        ;GIVE INDEX PULSE
035212  012777  000001  144430          MOV     #DMD,@RHMR      ;SET DIAGNOSTIC MODE
035220  052777  000004  144422          BIS     #MINX,@RHMR     ;SET INDEX
035226  042777  000004  144414          BIC     #MINX,@RHMR     ;CLEAR INDEX


                                        ;THESE ARE REGULAR SETUPS

035234  012777  177400  144360          MOV     #-256.,@RHWC    ;256 DATA  WORDS 4 HEADER WORDS
035242  012700  003126                  MOV     #REINTO,R0      ;THESE TWO INSTRUCTIONS GETS
035246  010077  144352                  MOV     R0,@RHBA        ;ADDR. OF WRFROM INTO R0 AND
                                                                ;BUS ADDRESS REGISTER
035252  012720  010000                  MOV     #FMT22,(R0)+    ;FORMAT=16 BIT WORDS
                                                                ;CYLINDER=0
035256  012720  012000                  MOV     #12000,(R0)+    ;TRACK=20 SECTOR=0 KEYS=0
035262  005020                          CLR     (R0)+           ;KEY1=0
035264  005020                          CLR     (R0)+           ;KEY2=0
035266  012705  000400                  MOV     #256.,R5        ;COUNTER
035272  012720  177777          1$:     MOV     #-1,(R0)+       ;MOVE ALL ONES FOR DATA
035276  005305                          DEC     R5
035300  001374                          BNE     1$              ;BRANCH IF DATA NOT COMPLETE
035302  012777  012000  144324          MOV     #12000,@RHDST   ;TRACK=20 SECTOR=0


035310  004767  002604                  JSR     PC,CHECKT               ;CHECK DVA, RDY, DPR, DRY


035314  013711  002044                  MOV     @#REFOR,@R1     ;GET READY FOR WRITE HEADER AND
                                                                ;DATA WITH 62 IN RHCS1
035320  005037  001772                  CLR     @#ERFLG$        ;CLEAR ERROR FLAG
035324  012777  010000  144306          MOV     #FMT22,@RHOF    ;FORMAT BIT=1 (16 BIT WORDS)
035332  005077  144304                  CLR     @RHCA           ;CYLINDER =0

                                        ;THE REGISTERS WILL BE SAVED IN REINTO BUFFER
035336  004037  040542                  JSR     R0,@#SAVER      ;SAVE
```

```
 9342  035342  001622                          RHWC                    ;FROM
 9343  035344  003126                          REINTO                  ;TO
 9344  035346  000023                          19.                     ;NUMBER SAVED
 9345
 9346                                          ;GO TO WRITE HEADER AND DATA
 9347
 9348  035350  013700  001650                  MOV     @#RHMR,R0       ;NOW R0 WAS MAINTENANCE REG. ADDR.
 9349  035354  012710  000001                  MOV     #DMD,@R0        ;SET DIAGNOSTIC MODE
 9350  035360  052777  000001  144242          BIS     #GO,@RHCS1      ;GO
 9351
 9352                                          ;CHANGE SAVED REGISTERS TO EXPECTED VALUE
 9353  035366  052737  140000  003134          BIS     #SC!TRE,@#REINTO+6 ;SAVED RHCS1
 9354  035374  012737  002000  003136          MOV     #IAE,@#REINTO+10 ;SAVED RHER1
 9355  035402  012737  012001  003140          MOV     #12001,@#REINTO+12;SAVED RHDST
 9356  035410  013737  002002  003152          MOV     @#ATTENT,@#REINTO+24 ;SAVED RHAS
 9357  035416  052737  000001  003154          BIS     #DMD,@#REINTO+26 ;SAVED RHMR
 9358  035424  052737  140000  003156          BIS     #ATA!ERR,@#REINTO+30 ;SAVED RHDS1
 9359
 9360                                          ;SAVE REGISTERS AGAIN SO THAT COMPARES CAN BE DONE
 9361  035432  004037  040542                  JSR     R0,@#SAVER      ;SAVE
 9362  035436  001622                          RHWC                    ;FROM
 9363  035440  002062                          WRFROM                  ;TO
 9364  035442  000023                          19.                     ;NUMBER OF REGISTERS SAVED
 9365
 9366                                          ;AS UPPER BYTE OF RHAS CAN BE CHANGING IN A DUAL PORT
 9367                                          ;OPERATION THE UPPER BYTE OF RHAS WILL BE SAVED AS IS
 9368                                          ;SO THAT THE COMPARES ARE ONLY VALID FOR THE LOWER BYTE
 9369  035444  113737  003153  002107          MOVB    @#REINTO+25,@#WRFROM+25;SAVE UPPER RHAS
 9370
 9371
 9372                                          ;COMPARE REGISTERS BEFORE READ IN PRESET COMMAND
 9373                                          ;WITH AFTER COMMAND
 9374  035452  004037  040736                  JSR     R0,@#COMPAR     ;COMPARE
 9375  035456  003126                          REINTO                  ;GOOD BUFFER
 9376  035460  002062                          WRFROM                  ;TEST BUFFER
 9377  035462  000021                          17.                     ;NUMBER OF REGISTERS
 9378  035464  035472                          2$                      ;RETURN FOR ERROR
 9379  035466  035472                          2$                      ;SAME
 9380  035470  035512                          3$                      ;RETURN FOR GOOD COMPARISON
 9381
 9382  035472  013705  044532          2$:     MOV     @#ERWORD,R5     ;GETTING READY TO INDEX
 9383  035476  060505                          ADD     R5,R5           ;DOUBLE ERROR WORD
 9384  035500  016537  001620  037534          MOV     RHWC-2(R5),@#REGADR ;FAILING REG. ADDRESS
 9385  035506  104001                          ERROR   1               ;FORCED IAE CAUSED IMPROPER
 9386                                                                  ;REGISTER CHANGE
 9387  035510  000207                          RTS     PC              ;RETURN FOR FURTHER COMPARISONS
 9388
 9389                                                                  ;NO ERRORS
 9390
 9391  035512  004737  040064          3$:     JSR     PC,@#CLDISK     ;CLEAR GO BIT
 9392
 9393
 9394                                          ;;**************************************************************************
 9395                                          ;*TEST 117       ERROR REGISTER 1- BIT #10 IAE
 9396
 9397                                          ;*      A WRITE HEADER AND DATA IS GIVEN TO SECTOR 22
```

```
                                        ;*      TRACK 0 CYLINDER 0
                                        ;*
                                        ;*      WORD COUNT IS SET TO 256.
                                        ;*
                                        ;*      AN INDEX PULSE IS GIVEN TO GET RHLA TO 0
                                        ;*
                                        ;*      IAE BIT SHOULD SET
                                        ;
                                        ;;*****************************************************************
035516  000004                        TST117: SCOPE
035520  012706  001000                        MOV     #STACK,SP       ;RESET STACK

035524  012737  000117  004172                MOV     #TTNO,@#TSTNM   ;THIS SAVES TEST NUMBER

035532  004737  040064                        JSR     PC,@#CLDISK     ;CLEAR REGISTERS AND SET UNIT NO.

                                              ;GIVE INDEX PULSE
035536  012777  000001  144104                MOV     #DMD,@RHMR      ;SET DIAGNOSTIC MODE
035544  052777  000004  144076                BIS     #MINX,@RHMR     ;SET INDEX
035552  042777  000004  144070                BIC     #MINX,@RHMR     ;CLEAR INDEX

                                              ;THESE ARE REGULAR SETUPS

035560  012777  177400  144034                MOV     #-256.,@RHWC    ;256 DATA WORDS 4 HEADER WORDS
035566  012700  002062                        MOV     #WRFROM,R0      ;THESE TWO INSTRUCTIONS GETS
035572  010077  144026                        MOV     R0,@RHBA        ;ADDR. OF WRFROM INTO R0 AND
                                                                      ;BUS ADDRESS REGISTER
035576  012720  010000                        MOV     #FMT22,(R0)+    ;FORMAT=16 BIT WORDS
                                                                      ;CYLINDER=0
035602  012720  000026                        MOV     #22.,(R0)+      ;TRACK=0, SECTOR=22, KEYS=0
035606  005020                              CLR     (R0)+           ;KEY1=0
035610  005020                              CLR     (R0)+           ;KEY2=0
035612  012705  000400                        MOV     #256.,R5        ;COUNTER
035616  012720  177777            1$:       MOV     #-1,(R0)+       ;MOVE ALL ONES FRO DATA
035622  005305                              DEC     R5
035624  001374                              BNE     1$              ;BRANCH IF DATA NOT COMPLETE
035626  012777  000026  144000                MOV     #22.,@RHDST     ;TRACK=0 SECTOR=22


035634  004767  002260                        JSR     PC,CHECKT               ;CHECK DVA, RDY, DPR, DRY

035640  013711  002040                        MOV     @#WRIFOR,@R1    ;GET READY FOR WRITE HEADER AND
                                                                      ;DATA WITH 62 IN RHCS1
035644  005037  001772                        CLR     @#ERFLG$        ;CLEAR ERROR FLAG
035650  012777  010000  143762                MOV     #FMT22,@RHOF    ;FORMA BIT=1 (16 BIT WORDS)
035656  005077  143760                        CLR     @RHCA           ;CYLINDER =0

                                              ;AS EXCEPTION IS ASSERTED BEFORE RUN IS
                                              ;LATCHED RHWC,RHBA,RHCS1,RHCS2 CANNOT BE CHECKED
                                              ;BECAUSE RHWC WILL VARY DEPENDING UPON GATE DELAYS
                                              ;ON DIFFERENT UNITS

                                              ;THE REGISTERS WILL BE SAVED IN REINTO BUFFER
```

```
9454  035662  004037  040542                  JSR     R0,@#SAVER      ;SAVE
9455  035666  001632                          RHER1                   ;FROM
9456  035670  003126                          REINTO                  ;TO
9457  035672  000015                          13.                     ;NUMBER SAVED
9458
9459                                          ;GO TO WRITE HEADER AND DATA
9460
9461  035674  013700  001650                  MOV     @#RHMR,R0       ;NOW R0 HAS MAINTENANCE REG. ADDR.
9462  035700  012710  000001                  MOV     #DMD,@R0        ;SET DIAGNOSTIC MODE
9463  035704  052777  000001  143716          BIS     #GO,@RHCS1      ;GO
9464
9465                                          ;CHANGE SAVED REGISTERS TO EXPECTED VALUE
9466  035712  012737  002000  003126          MOV     #IAE,@#REINTO   ;SAVED RHER1
9467  035720  012737  000027  003130          MOV     #23.,@#REINTO+2;SAVED RHDST
9468  035726  013737  002002  003142          MOV     @#ATTENT,@#REINTO+14 ;SAVED RHAS
9469  035734  052737  000001  003144          BIS     #DMD,@#REINTO+16 ;SAVED RHMR
9470  035742  052737  140000  003146          BIS     #ATA!ERR,@#REINTO+20 ;SAVED RHDS1
9471
9472                                          ;SAVE REGISTERS AGAIN SO THAT COMPARES CAN BE DONE
9473  035750  004037  040542                  JSR     R0,@#SAVER      ;SAVE
9474  035754  001632                          RHER1                   ;FROM
9475  035756  002062                          WRFROM                  ;TO
9476  035760  000015                          13.                     ;NUMBER OF REGISTERS SAVED
9477
9478                                          ;AS UPPER BYTE OF RHAS CAN BE CHANGING IN A DUAL PORT
9479                                          ;OPERATION THE UPPER BYTE OF RHAS WILL BE SAVED AS IS
9480                                          ;SO THAT THE COMPARES ARE ONLY VALID FOR THE LOWER BYTE
9481  035762  113737  003143  002077          MOVB    @#REINTO+15,@#WRFROM+15;SAVE UPPER RHAS
9482
9483
9484                                          ;COMPARE REGISTERS BEFORE READ IN PRESET COMMAND
9485                                          ;WITH AFTER COMMAND
9486  035770  004037  040736                  JSR     R0,@#COMPAR     ;COMPARE
9487  035774  003126                          REINTO                  ;GOOD BUFFER
9488  035776  002062                          WRFROM                  ;TEST BUFFER
9489  036000  000015                          13.                     ;NUMBER OF REGISTERS
9490  036002  036010                          2$                      ;RETURN FOR ERROR
9491  036004  036010                          2$                      ;SAME
9492  036006  036030                          3$                      ;RETURN FOR GOOD COMPARISON
9493
9494  036010  013705  044532          2$:     MOV     @#ERWORD,R5     ;GETTING READY TO INDEX
9495  036014  060505                          ADD     R5,R5           ;DOUBLE ERROR WORD
9496  036016  016537  001630  037534          MOV     RHER1-2(R5),@#REGADR ;FAILING REG. ADDRESS
9497  036024  104001                          ERROR   1               ;FORCED IAE CAUSED IMPROPER
9498                                                                  ;REGISTER CHANGE
9499  036026  000207                          RTS     PC              ;RETURN FOR FURTHER COMPARISONS
9500
9501                                                                  ;NO ERRORS
9502
9503  036030  004737  040064          3$:     JSR     PC,@#CLDISK     ;CLEAR GO BIT
9504
9505
9506
9507                                          ;;*****************************************************************
9508                                          ;*TEST 120       ERROR REGISTER 1- BIT #10 IAE
9509
```

```
                                        ;*      A WRITE DATA IS GIVEN TO SECTOR 0
                                        ;*      TRACK 0 CYLINDER 411
                                        ;*
                                        ;*      WORD COUNT IS SET TO 256.
                                        ;*      AN INDEX PULSE IS GIVEN TO GET RHLA TO 0
                                        ;*
                                        ;*
                                        ;*      IAE BIT SHOULD SET
                                        ;;*****************************************************************
036034  000004                    TST120: SCOPE
036036  012706  001000                    MOV     #STACK,SP       ;RESET STACK
036042  012737  000120  004172            MOV     #TTNO,@#TSTNM   ;THIS SAVES TEST NUMBER
036050  004737  040064                    JSR     PC,@#CLDISK     ;CLEAR REGISTERS AND SET UNIT NO.
                                        ;GIVE INDEX PULSE
036054  012777  000001  143566            MOV     #DMD,@RHMR      ;SET DIAGNOSTIC MODE
036062  052777  000004  143560            BIS     #MINX,@RHMR     ;SET INDEX
036070  042777  000004  143552            BIC     #MINX,@RHMR     ;CLEAR INDEX
                                        ;THESE ARE REGULAR SETUPS
036076  012777  177400  143516            MOV     #-256.,@RHWC    ;256 DATA WORDS 4 HEADER WORDS
036104  012700  002062                    MOV     #WRFROM,R0      ;THESE TWO INSTRUCTIONS GETS
036110  010077  143510                    MOV     R0,@RHBA        ;ADDR. OF WRFROM INTO R0 AND
                                                                ;BUS ADDRESS REGISTER
036114  012705  000400                    MOV     #256.,R5        ;COUNTER
036120  012720  177777            1$:     MOV     #-1,(R0)+       ;MOVE ALL ONES FRO DATA
036124  005305                    DEC     R5
036126  001374                    BNE     1$              ;BRANCH IF DATA NOT COMPLETE
036130  012777  000000  143476            MOV     #0.,@RHDST      ;TRACK=0 SECTOR=0

036136  004767  001756                    JSR     PC,CHECKT               ;CHECK DVA, RDY, DPR, DRY

036142  013711  002036                    MOV     @#WRIDAT,@R1    ;GET READY FOR WRITE
                                                                ;DATA WITH 60 IN RHCS1
036146  005037  001772                    CLR     @#ERFLG$        ;CLEAR ERROR FLAG
036152  012777  010000  143460            MOV     #FMT22,@RHOF    ;FORMA BIT=1 (16 BIT WORDS)
036160  012777  000633  143454            MOV     #411.,@RHCA     ;CYLINDER =411

                                        ;AS EXCEPTION IS ASSERTED BEFORE RUN IS
                                        ;LATCHED RHWC,RHBA,RHCS1,RHCS2 CANNOT BE CHECKED
                                        ;BECAUSE RHWC WILL VARY DEPENDING UPON GATE DELAYS
                                        ;ON DIFFERENT UNITS

                                        ;THE REGISTERS WILL BE SAVED IN REINTO BUFFER
036166  004037  040542                    JSR     R0,@#SAVER      ;SAVE
036172  001632                    RHER1                   ;FROM
036174  003126                    REINTO                  ;TO
036176  000015                    13.                     ;NUMBER SAVED
```

```
                                        ;GO TO WRITE HEADER AND DATA

036200  013700  001650                  MOV     @#RHMR,R0       ;NOW R0 HAS MAINTENANCE REG. ADDR.
036204  012710  000001                  MOV     #DMD,@R0        ;SET DIAGNOSTIC MODE
036210  052777  000001  143412          BIS     #GO,@RHCS1      ;GO

                                        ;CHANGE SAVED REGISTERS TO EXPECTED VALUE
036216  012737  002000  003126          MOV     #IAE,@#REINTO   ;SAVED RHER1
036224  012737  000001  003130          MOV     #1.,@#REINTO+2;SAVED RHDST
036232  013737  002002  003142          MOV     @#ATTENT,@#REINTO+14 ;SAVED RHAS
036240  052737  000001  003144          BIS     #DMD,@#REINTO+16 ;SAVED RHMR
036246  052737  140000  003146          BIS     #ATA!ERR,@#REINTO+20 ;SAVED RHDS1

                                        ;SAVE REGISTERS AGAIN SO THAT COMPARES CAN BE DONE
036254  004037  040542                  JSR     R0,@#SAVER      ;SAVE
036260  001632                          RHER1                   ;FROM
036262  002062                          WRFROM                  ;TO
036264  000015                          13.                     ;NUMBER OF REGISTERS SAVED

                                        ;AS UPPER BYTE OF RHAS CAN BE CHANGING IN A DUAL PORT
                                        ;OPERATION THE UPPER BYTE OF RHAS WILL BE SAVED AS IS
                                        ;SO THAT THE COMPARES ARE ONLY VALID FOR THE LOWER BYTE
036266  113737  003143  002077          MOVB    @#REINTO+15,@#WRFROM+15;SAVE UPPER RHAS

                                        ;COMPARE REGISTERS BEFORE READ IN PRESET COMMAND
                                        ;WITH AFTER COMMAND
036274  004037  040736                  JSR     R0,@#COMPAR     ;COMPARE
036300  003126                          REINTO                  ;GOOD BUFFER
036302  002062                          WRFROM                  ;TEST BUFFER
036304  000015                          13.                     ;NUMBER OF REGISTERS
036306  036314                          2$                      ;RETURN FOR ERROR
036310  036314                          2$                      ;SAME
036312  036334                          3$                      ;RETURN FOR GOOD COMPARISON

036314  013705  044532          2$:     MOV     @#ERWORD,R5     ;GETTING READY TO INDEX
036320  060505                          ADD     R5,R5           ;DOUBLE ERROR WORD
036322  016537  001630  037534          MOV     RHER1-2(R5),@#REGADR ;FAILING REG. ADDRESS
036330  104001                          ERROR   1               ;FORCED IAE CAUSED IMPROPER
                                                                ;REGISTER CHANGE
036332  000207                          RTS     PC              ;RETURN FOR FURTHER COMPARISONS

                                                                ;NO ERRORS

036334  004737  040064          3$:     JSR     PC,@#CLDISK     ;CLEAR GO BIT


                                        ;;*****************************************************************
                                        ;;*****************************************************************
                                        ;*TEST 121       END OF DRIVE

                                        ;*      THIS IS THE END OF TEST FOR ONE DRIVE
```

```
                                        ;*      IF THERE ARE MORE DRIVES THEN THE PROGRAM
                                        ;*      JUMPS TO TEST 5 FOR NEXT DRIVE TEST
                                        ;*      END PASS IS REACHED ONLY AFTER ALL DRIVES ARE COMPLETE
                                        ;;*****************************************************************
036340  000004                        TST121: SCOPE
036342  012767  000001  142642              MOV     #1,$TIMES       ;;DO 1 ITERATION
036350  012767  000000  141420              MOV     #0,PS           ;REINSTATE PS TO 0
036356  104400  036364                      TYPE    ,65$            ;;TYPE ASCIZ STRING
036362  000425                              BR      64$             ;;GET OVER THE ASCIZ
                                        ;;65$:  .ASCIZ  <15><12>/TOTAL ERRORS ON THIS PASS ON  UNIT NO. /
036436                                  64$:
036436  013746  001760                      MOV     @#UNIT,-(SP)    ;GET READY TO TYPE UNIT NUMBER
036442  104404                              TYPDS
036444  104400  036452                      TYPE    ,67$            ;;TYPE ASCIZ STRING
036450  000402                              BR      66$             ;;GET OVER THE ASCIZ
                                        ;;67$:  .ASCIZ  /= /
036456                                  66$:
036456  013746  001112                      MOV     @#$ERTTL,-(SP)  ;GET READY TO TYPE NUMBER OF ERRORS
036462  104404                              TYPDS
036464  005037  001112                      CLR     @#$ERTTL        ;CLEAR TOTAL NUMBER OF ERRORS
036470  005737  001766                      TST     @#SELECT        ;STARTING FROM 200 ?
036474  001415                              BEQ     3$              ;BRANCH IF YES
036476  005067  142400                      CLR     $TSTNM          ;CLEAR TEST NUMBER
036502  005237  001100                      INC     @#$PASS         ;INCREASE PASS COUNT
036506  104400  036671                      TYPE    ,$ENDMG         ;TYPE END PASS #
036512  013746  001100                      MOV     @#$PASS,-(SP)
036516  104404                              TYPDS
036520  104400  036666                      TYPE    ,$ENULL
036524  000137  007120                      JMP     @#TST5          ;JUMP TEST 5
036530  005337  001762              3$:     DEC     @#NOUNITS       ;NO. OF UNITS PRESENT DECREMENT
036534  001413                              BEQ     $EOP            ;BRANCH IF ALL DRIVES COMPLETE
036536  013700  001760                      MOV     @#UNIT,R0       ;UNIT UNDER TEST
036542  012701  001740                      MOV     #UNITS,R1       ;TABLE
036546  022100                      1$:     CMP     (R1)+,R0        ;IS THIS UNIT JUST TESTED
036550  001401                              BEQ     2$              ;BRANCH IF YES
036552  000775                              BR      1$              ;BRANCH IF NO
036554  011137  001760              2$:     MOV     (R1),@#UNIT     ;THIS IS NEXT UNIT
036560  000137  007120                      JMP     @#TST5          ;GO FOR NEXT TESTS.
```

```
                                        .SBTTL  END OF PASS ROUTINE

                                        ;;**********************************************************************
                                        ;*INCREMENT THE PASS NUMBER ($PASS)
                                        ;*TYPE "END PASS #XXXXX" (WHERE XXXXX IS A DECIMAL NUMBER)
                                        ;*IF THERES A MONITOR GO TO IT
                                        ;*IF THERE ISN'T JUMP TO TST1

036564                                  $EOP:
036564  000004                                SCOPE
036566  005067  142310                       CLR     $TSTNM          ;;ZERO THE TEST NUMBER
036572  005067  142414                       CLR     $TIMES          ;;ZERO THE NUMBER OF ITERATIONS
036576  005267  142276                       INC     $PASS           ;;INCREMENT THE PASS NUMBER
036602  042767  100000  142270               BIC     #100000,$PASS   ;;DON'T ALLOW A NEG. NUMBER
036610  005327                             $EOPCT: DEC     (PC)+           ;;LOOP?
036612  000001                                     .WORD   1
036614  003022                                     BGT     $DOAGN          ;;YES
036616  012737                                     MOV     (PC)+,@(PC)+    ;;RESTORE COUNTER
036620  000001                             $ENDCT: .WORD   1
036622  036612                                     $EOPCT
036624  104400  036671                       TYPE    ,$ENDMG         ;;TYPE "END PASS #"
036630  016746  142244                       MOV     $PASS,-(SP)     ;;SAVE $PASS FOR TYPEOUT
036634  104404                             TYPDS                   ;;GO TYPE--DECIMAL ASCII WITH SIGN
036636  104400  036666                     TYPE    ,$ENULL         ;;TYPE A NULL CHARACTER
036642  013700  000042                     $GET42: MOV     @#42,R0         ;;GET MONITOR ADDRESS
036646  001405                             BEQ     $DOAGN          ;;BRANCH IF NO MONITOR
036650  000005                             RESET                   ;;CLEAR THE WORLD
036652  004710                             $ENDAD: JSR     PC,(R0)         ;;GO TO MONITOR
036654  000240                             NOP                     ;;SAVE ROOM
036656  000240                             NOP                     ;;FOR
036660  000240                             NOP                     ;;ACT11
036662                                  $DOAGN:
036662  000137                             JMP     @(PC)+          ;;RETURN
036664  005414                          $RTNAD: .WORD   TST1
036666     377     377     000          $ENULL: .BYTE   -1,-1,0         ;;NULL CHARACTER STRING
036671     015  042412  042116          $ENDMG: .ASCIZ  <15><12>/END PASS #/
036676  050040  051501  020123
036704  000043

                                                .SBTTL  SUBROUTINES
                                        ;HERE IS A DETAILED EXPLAINATION OF HOW THE LOOP ON ERROR WORKS.
                                        ;ON HITTING AN ERROR IF THE LOOP ON ERROR SWITCH IS SET, THE
                                        ;PROGRAM GOES BACK - USUALLY BACK TO THE BEGINNING OF THE TEST.

                                        ;WHEN THIS OPERATOR SELECTABLE SCOPE LOOP IS USED THEN THE POINT
                                        ;THE PROGRAM GOES BACK TO CAN BE CHANGED.
                                        ;THE RESTRICTIONS TO THE POINT WHERE THE PROGRAM CAN GO ARE: -
                                        ;1.  IT MUST BE WITHIN THE TEST UNDER CONSIDERATION
                                        ;2.  LOOP ON ERROR SWITCH MUST BE SET
                                        ;3.  THE ERROR MUST OCCUR WITHIN THE TEST UNDER CONSIDERATION
                                        ;IF THE ERROR DOES NOT OCCUR WITHIN THE TEST UNDER CONSIDERATION
                                        ;THE PROGRAM WILL REVERT TO NORMAL OPERATION.  HOWEVER, IF LOOP ON
                                        ;TEST SWITCH IS SET AND THIS OPERATOR SELECTABLE SCOPE LOOP IS USED
                                        ;THEN THE PROGRAM WILL LOOP BACK TO THE SELECTED POINT WHEN IT
```

```
                                        ;COMES TO THE END OF THE TEST UNDER CONSIDERATION.
                                        ;
                                        ;AFTER LOOPING FOR SOME TIME IF THE LOOP SWITCH IS PUT DOWN THEN
                                        ;NORMAL OPERATION WILL CONTINUE.

036706  000000                  TESTAD: 0                       ;FIRST ADDRESS OF TEST
036710                          OPERSEL:
036710  005067  141062                  CLR     PS              ;MAKE PROCESSOR STATUS ZERO
036714  104400  036722                  TYPE    ,65$            ;;TYPE ASCIZ STRING
036720  000421                          BR      64$             ;;GET OVER THE ASCIZ
                                ;;65$:  .ASCIZ  <15><12>/THE PROGRAM WAS IN TEST NUMBER /
036764                          64$:
036764  013746  004172                  MOV     @#TSTNM,-(SP)   ;GET READY TO TYPE TEST
036770  104401                          TYPOC                   ;NUMBER
036772  104400  037000                  TYPE    ,67$            ;;TYPE ASCIZ STRING
036776  000414                          BR      66$             ;;GET OVER THE ASCIZ
                                ;;67$:  .ASCIZ  <15><12>/THE LOOP BACK PC WAS /
037030                          66$:
037030  013746  001110                  MOV     @#$LPERR,-(SP)  ;GET READY TO TYPE LOOP BACK PC
037034  104401                          TYPOC
037036  104400  001223                  TYPE    ,$CRLF
037042  104400  037050                  TYPE    ,69$            ;;TYPE ASCIZ STRING
037046  000430                          BR      68$             ;;GET OVER THE ASCIZ
                                ;;69$:  .ASCIZ  <15><12>/SET SWITCH FOR LOOP ON ERROR OR LOOP ON TEST/
037130                          68$:
037130  104400  037136                  TYPE    ,71$            ;;TYPE ASCIZ STRING
037134  000430                          BR      70$             ;;GET OVER THE ASCIZ
                                ;;71$:  .ASCIZ  <15><12>/TYPE THE FIRST PC OF THE TEST TO BE LOOPED ON/
037216                          70$:
037216  104400  037224                  TYPE    ,73$            ;;TYPE ASCIZ STRING
037222  000422                          BR      72$             ;;GET OVER THE ASCIZ
                                ;;73$:  .ASCIZ  <15><12>/ FOLLOWED BY A CARRIAGE RETURN /<15><12>
037270                          72$:
037270  104407                          RDOCT
037272  062716  000002                  ADD     #2,(SP)         ;GET LPADR
037276  012637  001106                  MOV     (SP)+,@#$LPADR
037302  104400  037310                  TYPE    ,75$            ;;TYPE ASCIZ STRING
037306  000417                          BR      74$             ;;GET OVER THE ASCIZ
                                ;;75$:  .ASCIZ  <15><12>/TYPE THE PC WHERE YOU WANT/
037346                          74$:
037346  104400  037354                  TYPE    ,77$            ;;TYPE ASCIZ STRING
037352  000440                          BR      76$             ;;GET OVER THE ASCIZ
                                ;;77$:  .ASCIZ  <15><12>/ THE PROGRAM TO LOOP BACK TO FOLLOWED BY A CARRIAGE RETURN /<15
037454                          76$:
037454  104407                          RDOCT
037456  012637  001110                  MOV     (SP)+,@#$LPERR  ;GET LPERR
037462  013746  001106                  MOV     @#$LPADR,-(SP)
037466  000002                          RTI


                                        .SBTTL  SAVE REGISTERS ROUTINE
                                        ;THIS SAVES THE CONTENTS OF ALL HARDWARE REGISTERS
                                        ;IN MEMORY LOCATIONS TAGED FROM "WC" TO "EC2"
```

```
                                        ;THIS IS DONE SO THAT COMPARES ARE DONE WITH SAVED LOCATIONS
                                        ;AND NOT THE REGISTERS THEMSELVES.   THIS WILL MAKE
                                        ;ERROR PRINTOUTS FOR GOOD AND BAD DATA ALWAYS DIFFRENT

037470                          PUTREG:
037470  010046                        MOV     R0,-(SP)        ;;PUSH R0 ON STACK
037472  010146                        MOV     R1,-(SP)        ;;PUSH R1 ON STACK
037474  010246                        MOV     R2,-(SP)        ;;PUSH R2 ON STACK
037476  012700  001622                MOV     #RHWC,R0        ;STARTING ADDRESS OF REG
037502  012701  001672                MOV     #WC,R1          ;STARTING ADDRESS OF WERE SAVED
037506  012702  000023                MOV     #RHCC-RHWC+2/2,R2 ;NUMBER OF REG. INTO R2
037512  013021                10$:    MOV     @(R0)+,(R1)+    ;SAVE HARDWARE REG.
037514  005302                        DEC     R2
037516  001375                        BNE     10$
037520  012602                        MOV     (SP)+,R2        ;;POP STACK INTO R2
037522  012601                        MOV     (SP)+,R1        ;;POP STACK INTO R1
037524  012600                        MOV     (SP)+,R0        ;;POP STACK INTO R0
037526  000207                        RTS     PC


                                        .SBTTL  FLOAT 1 AND 0
                                ;FLOAT A ONE AND A ZERO THRU A DESIGNATED REGISTER
                                ;ABSOLUTE ADDRESS OF REG. UNDER TEST IS IN R4

037530  000000                  MASK:   0                       ;BITS UNDER TEST
037532  000000                  LERR:   0                       ;ERROR HLT ADDRESS
037534  000000                  REGADR: 0

037536  012567  177766          BITST:  MOV     (R5)+,  MASK    ;FETCH DATA MASK
037542  012504                          MOV     (R5)+,  R4      ;GET ADDRESS OF REG. UNDER TEST
037544  010467  177764                  MOV     R4,     REGADR
037550  010567  177756                  MOV     R5,     LERR    ;GET ERROR RETURN ADDR.
037554  062705  000004                  ADD     #4,     R5      ;MODIFY RETURN ADDR. TO JUMP OVER RTS
037560  012703  000001                  MOV     #1,     R3      ;INITIALIZE DATA PATTERN
037564  004767  000016          BLT1:   JSR     PC,     BLT2    ;OUTPUT FLOATING ZERO
037570  004767  000012                  JSR     PC,     BLT2    ;OUTPUT FLOATING ONE
037574  000241                          CLC
037576  006103                          ROL     R3              ;SHIFT PATTERN
037600  005703                          TST     R3
037602  001370                          BNE     BLT1            ;BRANCH IF NOT COMPLETE
037604  000205                          RTS     R5              ;RETURN TO TEST
037606  005103                  BLT2:   COM     R3              ;COMPLEMENT PATTERN
037610  012737  037616  040044        BLT3:   MOV     #BLT3,  @#LAD   ;SET SCOPE LOOP
037616  032737  001000  001140                BIT     #SW09,@#SWR     ;LOOP ON ERROR
037624  001411                          BEQ     4$              ;BRANCH IF NO
037626  105737  001103                  TSTB    @#$ERFLG        ;ANY ERRORS
037632  001406                          BEQ     4$      ;BRANCH IF NO
037634  000005                          RESET                   ;START WITH AN INIT
037636  013777  001760  141762        MOV     @#UNIT,@RHCS2   ;SET UNIT NUMBER UNDER TEST
037644  004737  052030                JSR     PC,@#$TKINT     ;INITILIZE TK
                                                                ;INIT FOR SCOPING LOOPS
037650  010337  001124          4$:     MOV     R3,@#$GDDAT     ;STORE GOOD DATA
037654  005137  037530                  COM     @#MASK          ;AND MASK WITH PATTERN
```

```
 9829  037660  043737  037530  001124          BIC     @#MASK, @#$GDDAT        ;CLEAR THE REST
 9830  037666  005137  037530                  COM     @#MASK          ;RESTORE MASK
 9831  037672  013714  001124                  MOV     @#$GDDAT,(R4)   ;OUTPUT TO REGISTER
 9832  037676  011437  001126                  MOV     (R4),@#$BDDAT   ;INPUT FROM REGISTER
 9833  037702  005137  037530                  COM     @#MASK
 9834  037706  043737  037530  001126          BIC     @#MASK,@#$BDDAT ;AND MASK OUT RECEIVED DATA
 9835  037714  005137  037530                  COM     @#MASK          ;RESTORE MASK
 9836  037720  023737  001124  001126          CMP     @#$GDDAT,@#$BDDAT       ;IS DATA CORRECT
 9837  037726  001424                          BEQ     1$              ;BRANCH IF GOOD
 9838  037730  011437  001126                  MOV     (R4),@#$BDDAT
 9839  037734  023704  001630                  CMP     @#RHCS1,R4      ;REGISTER UNDER TEST RHCS1?
 9840  037740  001004                          BNE     2$              ;BRANCH IF NOT
 9841  037742  052737  004200  001124          BIS     #RDY!DVA,@#$GDDAT;SET RDY AND DVA
 9842  037750  000410                          BR      3$
 9843  037752  023704  001626          2$:     CMP     @#RHCS2,R4      ;REGISTER UNDER TEST RHCS2?
 9844  037756  001005                          BNE     3$              ;BRANCH IF NOT
 9845  037760  011446                          MOV     @R4,-(SP)       ;GET RHCS2
 9846  037762  042716  177477                  BIC     #^C<IR!OR>,(SP) ;KEEP IR AND OR BIT
 9847  037766  052637  001124                  BIS     (SP)+,@#$GDDAT  ;SET IR OR BITS IF NEEDED
 9848  037772  004777  177534          3$:     JSR     PC,     @LERR   ;GO TO REPORT ERROR
 9849  037776  000240                          NOP                     ;REPLACE BY 104420 FOR LOCAL SCOPE LOOP
 9850  040000  000207          1$:             RTS     PC
 9851                                          .SBTTL  CLEAR MEMORY ROUTINE
 9852
 9853
 9854
 9855                          ;       THIS CLEARS ANY BLOCK OF MEMORY
 9856                          ;       FILLING IT WITH ANY DATA
 9857                          ;
 9858                          ;       CALL
 9859                          ;       JSR     R0,CLAREA
 9860                          ;       X                       ;STARTING ADDRESS OF BLOCK
 9861                          ;       Y
 9862                          ;       Z                       ;DATA TO BE FILLED
 9863                          ;R1 WILL HAVE STARTING ADDRESS OF BLOCK TO BE FILLED
 9864                          ;R2 AFTER SUBTRACTION WILL HAVE TWICE NUMBER OF LOCATIONS
 9865                          ;R3 WILL HAVE DATA TO BE FILLED
 9866                          ;TO AVOID DIVIDE ROUTINE TWO DECREMENT R2 WILL BE USED
 9867  040002                  CLAREA:
 9868  040002  010146                  MOV     R1,-(SP)        ;;PUSH R1 ON STACK
 9869  040004  010246                  MOV     R2,-(SP)        ;;PUSH R2 ON STACK
 9870  040006  010346                  MOV     R3,-(SP)        ;;PUSH R3 ON STACK
 9871  040010  012001                  MOV     (R0)+,R1        ;FROM
 9872  040012  012002                  MOV     (R0)+,R2        ;TO
 9873  040014  012003                  MOV     (R0)+,R3        ;DATA
 9874  040016  160102                  SUB     R1,R2           ;NO. OF LOCATIONS MINUS TWO
 9875  040020  062702  000002          ADD     #2,R2           ;GET TWICE NO OF LOCATIONS
 9876  040024  010321          1$:     MOV     R3,(R1)+        ;MOVE IN DATA
 9877  040026  005302                  DEC     R2
 9878  040030  005302                  DEC     R2
 9879  040032  001374                  BNE     1$              ;BRANCH IF NOT COMPLETE
 9880  040034  012603                  MOV     (SP)+,R3        ;;POP STACK INTO R3
 9881  040036  012602                  MOV     (SP)+,R2        ;;POP STACK INTO R2
 9882  040040  012601                  MOV     (SP)+,R1        ;;POP STACK INTO R1
 9883  040042  000200                  RTS     R0              ;RETURN
```

```
                                                .SBTTL  LOCAL TRAPS
040044  000000                          LAD:    0

040046  032737  001000  001140  T.SCOP: BIT     #SW09,  @#SWR
040054  001402                          BEQ     1$
040056  013716  040044                  MOV     @#LAD,  (SP)
040062  000002                  1$:     RTI

                                        ;EXAMPLE OF THE USE OF THE ABOVE
                                        ;THIS WILL LOOP BETWEEN X: AND SCOP1 PROVIDED THERE IS NO "NEWTST"
                                        ;MOV    #X,     @#LAD
                                        ;X:     ---     ---
                                        ;       ---     ---
                                        ;       ---     ---
                                        ;       SCOP1

                                                .SBTTL  CLEAD DISK ROUTINE

040064  013701  001630          CLDISK: MOV     @#RHCS1,        R1      ;R1 WILL BE CONTROL AND STATUS1
040070  013702  001626                  MOV     @#RHCS2,        R2      ;R2 WILL BE CONTROL AND STATUS2
040074  013703  001652                  MOV     @#RHDS1,        R3      ;R3 WILL BE DISK STATUS REGISTER1
040100  013704  001632                  MOV     @#RHER1,        R4      ;R4 WILL BE ERROR REGISTER #1

040104  012712  000040                  MOV     #CLR,@R2                ;CLEAR ALL REG.
040110  013712  001760                  MOV     @#UNIT,@R2      ;REINSTATE UNIT NO.
040114  005011                          CLR     @R1                     ;CLEAR FUNCTION BITS
040116  000207                          RTS     PC

                                                .SBTTL  CHECK DISK STATUS ROUTINE

                                        ;THIS CHECKS DEVICE AVAILABLE (DVA) AND READY (RDY) IN RHCS1
                                        ;AND CHECKS MEDIUM ON LINE (MOL), DEVICE PRESENT (DPR), DEVICE READY (DRY) IN RHDS1

040120  011637  002000          CHECKT: MOV     (SP),@#PCJSR    ;SAVE PC OF JSR+4
040124  162737  000004  002000          SUB     #4,@#PCJSR      ;GET PC OF JSR
040132  004737  037470                  JSR     PC,@#PUTREG     ;SAVE REGISTERS
040136  022737  004200  001700          CMP     #DVA!RDY,@#CS1  ;RHCS1 SHOULD HAVE DEVICE AVAILABLE
                                                                ;AND BE READY
040144  001423                          BEQ     3$              ;BRANCH IF GOOD
040146  032737  004000  001700          BIT     #DVA,   @#CS1   ;BAD SO TEST DEVICE AVAILABLE
040154  001004                          BNE     1$              ;BRANCH IF DVA THERE
040156  010137  001122                  MOV     R1,@#$BDADR     ;ADDRESS OF BAD REGISTER (RHCS1)
040162  104026                          ERROR   26              ;RHCS1 DID NOT HAVE DEVICE
                                                                ;AVAILABLE RIGHT AT THE START
040164  000413                          BR      3$              ;BRANCH TO NEXT COMPARE
040166  032737  000200  001700  1$:     BIT     #RDY,   @#CS1   ;TEST READY
040174  001003                          BNE     2$              ;IF RDY THERE BRANCH
040176  010137  001122                  MOV     R1,@#$BDADR     ;ADDRESS OF BAD REGISTER (RHCS1)
040202  104026                          ERROR   26              ;RHCS1 DID NOT HAVE READY
                                                                ;RIGHT AT THE START
040204  000403                  2$:     BR      3$              ;BRANCH TO NEXT COMPARE
040206  010137  001122                  MOV     R1,@#$BDADR     ;ADDRESS OF BAD REGISTER (RHCS1)
040212  104026                          ERROR   26              ;RHCS1 HAD SOME BITS OTHER
```

```
                                                                ;THAN DVA AND RDY SET
                                                                ;ALL OTHER BITS SHOULD BE 0
040214  013746  001722          3$:   MOV     @#DS1,-(SP)       ;GET RHDS1
040220  042716  001100                BIC     #VV!PROG,(SP)     ;CLEAR VV AND PROGRAMABLE BIT
040224  022726  000600                CMP     #DPR!DRY,(SP)+;RHDS1 SHOULD HAVE THESE SET
040230  001423                      BEQ     7$                ;BRANCH IF GOOD
040232  032737  000400  001722  4$:   BIT     #DPR,   @#DS1     ;TEST DRIVE PRESENT
040240  001004                      BNE     5$                ;IF MOL WAS THERE SO BRANCH
040242  010337  001122              MOV     R3,@#$BDADR       ;ADDRESS OF BAD REGISTER (RHDS1)
040246  104026                      ERROR   26                ;RHDS1 DOES NOT HAVE DPR
040250  000413                      BR      7$                ;BRANCH OUT
040252  032737  000200  001722  5$:   BIT     #DRY,   @#DS1     ;TEST DRIVE READY
040260  001004                      BNE     6$                ;IF DPR WAS THERE SO BRANCH
040262  010337  001122              MOV     R3,@#$BDADR       ;ADDRESS OF BAD REGISTER (RHDS1)
040266  104026                      ERROR   26                ;RHDS1 DOES NOT HAVE DRY
040270  000403                      BR      7$                ;BRANCH OUT
040272  010337  001122          6$:   MOV     R3,@#$BDADR       ;ADDRESS OF BAD REGISTER (RHDS1)
040276  104026                      ERROR   26                ;RHDS1 HAS SOME BITS OTHER
                                                                ;THAN MOL, DRY, DPR, SET
                                                                ;ALL OTHER BITS SHOULD BE 0
040300  000207          7$:   RTS     PC                              ;RETURN TO TEST NO.


                        ;THIS CHECKS DEVICE AVAILABLE (DVA) AND READY (RDY) IN RHCS1
                        ;AND CHECKS MEDIUM ON LINE (MOL), DEVICE PRESENT (DPR), DEVICE READY (DRY) IN RHDS1

040302  011637  002000  CHECKE: MOV     (SP),@#PCJSR      ;SAVE PC OF JSR+4
040306  162737  000004  002000  SUB     #4,@#PCJSR        ;GET PC OF JSR
040314  004737  037470          JSR     PC,@#PUTREG       ;SAVE REGISTERS
040320  032737  000200  001700  BIT     #RDY,@#CS1        ;RHCS1 SHOULD HAVE DEVICE AVAILABLE
                                                          ;AND BE READY
040326  001004                  BNE     1$                ;BRANCH IF GOOD
040330  010137  001122          MOV     R1,@#$BDADR       ;FAILING REGISTER
040334  104026                  ERROR   26                ;RHCS1 IS IN ERROR
                                                          ;DOES NOT HAVE DVA, RDY
040336  000427                  BR      4$                ;BRANCH
040340  032737  004000  001700  1$:   BIT     #DVA,@#CS1        ;RHCS1 SHOULD HAVE DEVICE AVAILABLE
                                                          ;AND BE READY
040346  001004                  BNE     2$                ;BRANCH IF GOOD
040350  010137  001122          MOV     R1,@#$BDADR       ;FAILING REGISTER
040354  104026                  ERROR   26                ;RHCS1 IS IN ERROR
                                                          ;DOES NOT HAVE DVA, RDY
040356  000417                  BR      4$                ;BRANCH OUT
040360  032737  000200  001722  2$:   BIT     #DRY,@#DS1        ;RHDS1 SHOULD HAVE DPR,DRY
040366  001004                  BNE     3$                ;BRANCH IF THERE
040370  010337  001122          MOV     R3,@#$BDADR       ;FAILING REGISTER RHDS1
040374  104026                  ERROR   26                ;RHDS1 DOES NOT HAVE DPR,DRY
040376  000407                  BR      4$                ;BRANCH OUT
040400  032737  000400  001722  3$:   BIT     #DPR,@#DS1        ;RHDS1 SHOULD HAVE DPR,DRY
040406  001003                  BNE     4$                ;BRANCH IF THERE
040410  010337  001122          MOV     R3,@#$BDADR       ;FAILING REGISTER RHDS1
040414  104026                  ERROR   26                ;RHDS1 DOES NOT HAVE DPR,DRY
040416  000207          4$:   RTS     PC
```

```
                                                ;       WAIT LOOP
                                                ;ONE  LOOP OR ONE COUNT = 5.15 MICROSEC WITH BIPOLAR MEMORY (MIN)
                                                ;ONE LOOP OR ONE COUNT = 11.86 MICROSEC WITH CORE (MIN)
                                                ;WITH CORE ERROR IS INDICATED AFTER ABOUT 650 MILLISEC (MIN)
040420  177777                          TIMCNT: 177777                  ;WAITING COUNT
040422  010046                          WAIT.T: MOV     R0,-(SP)                ;SAVE R0
040424  016600  000002                          MOV     2(SP),R0                ;GET ADRESS OF REG. ADDERSS
040430  010037  001204                          MOV     R0,@#$TMP3              ;WAT PC+2 IN $TMP3
040434  162737  000002  001204                  SUB     #2,@#$TMP3              ;WAT PC FOR TYPEOUT
040442  012037  001176                          MOV     (R0)+,@#$TMP0           ;WAIT REGISTER ADDRESS
040446  012037  001200                          MOV     (R0)+,@#$TMP1           ;WAIT ON BIT
040452  010066  000002                          MOV     R0,2(SP)                ;RESTORE RETURN ON STACK
040456  012600                                  MOV     (SP)+,R0                ;RESTORE R0
040460  013737  040420  001202                  MOV     @#TIMCNT,@#$TMP2        ;TEMPORARY COUNT
040466  033777  001200  140502  1$:     BIT     @#$TMP1,@$TMP0  ;IS REQUIRED BIT THERE?
040474  001021                                  BNE     2$              ;BRANCH IF YES
040476  005337  001202                          DEC     @#$TMP2         ;COUNT
040502  001371                                  BNE     1$              ;BRANCH IF NOT TIME UP
040504  013737  040420  001202                  MOV     @#TIMCNT,@#$TMP2        ;TEMPORARY COUNT
040512  033777  001200  140456  3$:     BIT     @#$TMP1,@$TMP0  ;IS REQUIRED BIT THERE?
040520  001007                                  BNE     2$              ;BRANCH IF YES
040522  005337  001202                          DEC     @#$TMP2         ;COUNT
040526  001371                                  BNE     3$              ;BRANCH IF NOT TIME UP
040530  017737  140442  001126                  MOV     @$TMP0,@#$BDDAT ;REGISTER CONTENTS
040536  104016                                  ERROR   16              ;WAITED ON BIT FAILED TO SET
040540  000002                          2$:     RTI
                                                ;       CALL FOR THE ABOVE WAITLOOP IS
                                                ;
                                                ;       MOV     @A,@#X$         ;A CONTAINS REGISTER ADDRESS
                                                ;       -       -       -       ;HENCE X$ WILL HAVE ABSOLUTE REG. ADR.
                                                ;       -       -       -
                                                ;       -       -       -
                                                ;       WAT
                                                ;X$:    0                       ;ABSOLUTE REG. ADDRESS UNDER WAIT
                                                ;       .WORD   0               ;BIT WAITED FOR
                                                ;                               ;CONTINUE


                                                ;THIS IS A SUBROUTINE TO SAVE REGISTERS
                                                ;IN THE REGISTER TABLE TO ANY LOCATION
                                                .SBTTL  SAVE ROUTINE
                                                ;THE CALL IS
                                                ;JSR     R0,@#SAVER
                                                ;FROM
                                                ;TO
                                                ;NUMBER OF WORDS SAVED

040542                          SAVER:
040542  010146                          MOV     R1,-(SP)        ;;PUSH R1 ON STACK
040544  010246                          MOV     R2,-(SP)        ;;PUSH R2 ON STACK
040546  010346                          MOV     R3,-(SP)        ;;PUSH R3 ON STACK
```

```
10052  040550  012001                        MOV     (R0)+,R1        ;FROM
10053  040552  012002                        MOV     (R0)+,R2        ;TO
10054  040554  012003                        MOV     (R0)+,R3        ;NUMBER
10055  040556  013122                1$:     MOV     @(R1)+,(R2)+    ;SAVE REGISTER:CONTENTS
10056  040560  005303                        DEC     R3              ;COUNT
10057  040562  001375                        BNE     1$              ;BRANCH IF NOT DONE
10058  040564  012603                        MOV     (SP)+,R3        ;;POP STACK INTO R3
10059  040566  012602                        MOV     (SP)+,R2        ;;POP STACK INTO R2
10060  040570  012601                        MOV     (SP)+,R1        ;;POP STACK INTO R1
10061  040572  000200                        RTS     R0
10062
10063
10064
10065
10066                                        .SBTTL  WRITE CHECK ROUTINE
10067                                        ;THIS IS A SUBROUTINE TO DO WRITE CHECK HEADER AND DATA
10068                                        ;CYLINDER 0, TRACK 1, SECTOR 1, KEYS 0
10069
10070                                        ;THESE ARE TO SET UP FOR DISKLESS USE ONLY
10071  040574  012737  010000  044412  WRCHHD: MOV     #FMT22,@#CYL    ;CYLINDER 0 FORMAT 16 BIT WORDS
10072  040602  112737  000001  044415        MOVB    #1,@#SECOTR+1   ;TRACK=1
10073  040610  112737  000001  044414        MOVB    #1,@#SECOTR     ;SECTOR=1
10074  040616  005037  044416                CLR     @#KEY1          ;KEY1=0
10075  040622  005037  044420                CLR     @#KEY2          ;KEY2=0
10076  040626  012767  000044  003636        MOV     #36.,DAWORD     ;NO OF DATA WORDS
10077  040634  005037  044422                CLR     @#X             ;THIS IS A READ OPERATION
10078  040640  004537  041242                JSR     R5,@#CRC        ;GO TO CALCULATE CRC
10079  040644  044412                        CYL
10080  040646  046312                        WCRC
10081
10082                                        ;THESE ARE REGULAR SETUPS
10083
10084  040650  004737  040064                JSR     PC,@#CLDISK     ;SET UP GENERAL REGISTERS
10085                                                                 ;AND CLEAR DISK REGISTERS
10086  040654  012777  177730  140740        MOV     #-40.,@RHWC     ;36 DATA WORDS 4 HEADER WORDS
10087  040662  012777  003126  140734        MOV     #REINTO,@RHBA   ;STARTING ADDRESS OF READ BUFFER
10088  040670  112746  000001                MOVB    #1,-(SP)        ;SECTOR=1
10089  040674  112766  000001  000001        MOVB    #1,1(SP)        ;TRACK=1 IN UPPER BYTE
10090  040702  012677  140726                MOV     (SP)+,@RHDST    ;TRACK=1, SECTOR=1 IN RHDST
10091  040706  012777  014000  140724        MOV     #FMT22!ECI,@RHOF ;16 BIT WORDS
10092                                                                 ;ECC CORRECTION INHIBIT BECAUSE
10093                                                                 ;ECC LOGIC IS NOT CHECKED YET
10094  040714  005077  140722                CLR     @RHCA           ;CYLINDER=0
10095  040720  004737  040120                JSR     PC,@#CHECKT     ;CHECK FOR DVA,RDY,MOL,DPR,DRY
10096  040724  013711  002034                MOV     @#WRCHDT,@R1    ;WRITE CHECK HEADER AND DATA=52
10097                                                                 ;INTO RHCS1
10098  040730  004737  044302                JSR     PC,@#COMHD      ;WRITE CHECK HEADER AND DATA
10099                                                                 ;SAME AS READ HEADER AND DATA
10100
10101  040734  000207                        RTS     PC              ;RETURN TO WRITE CHECK TEST
10102                                        .SBTTL  COMPARE ROUTINE
10103                                ;THIS IS A SUBROUTINE TO COMPARE TWO BLOCKS IN MEMORY
10104                                ;R1 HAS GOOD DATA BUFFER ADDRESS
10105                                ;R2 HAS TEST DATA BUFFER ADDRESS
10106                                ;$TMP0 HAS ADDRESS OF RETURN ON ERROR TO PRINT HEADER
10107                                ;$TMP1 HAS ADDRESS OF RETURN ON ERROR TO PRINT DATA
```

```
10108                                   ;R3 HAS NUMBER OF WORDS TO BE COMPARED
10109                                   ;R4 HAS ONE MORE THAN NUMBER OF WORDS TO BE COMPARED
10110
10111  040736                           COMPAR:
10112  040736  010146                          MOV     R1,-(SP)        ;;PUSH R1 ON STACK
10113  040740  010246                          MOV     R2,-(SP)        ;;PUSH R2 ON STACK
10114  040742  010346                          MOV     R3,-(SP)        ;;PUSH R3 ON STACK
10115  040744  010446                          MOV     R4,-(SP)        ;;PUSH R4 ON STACK
10116  040746  010546                          MOV     R5,-(SP)        ;;PUSH R5 ON STACK
10117  040750  012001                          MOV     (R0)+,R1        ;ADDRESS OF GOOD DATA BUFFER
10118  040752  012002                          MOV     (R0)+,R2        ;ADDRESS OF TEST DATA BUFFER
10119  040754  012003                          MOV     (R0)+,R3        ;NO OF WORDS TO BE COMPARED
10120  040756  012067  140214                  MOV     (R0)+,$TMP0     ;RETURN ON ERROR TO PRINT HEADER
10121  040762  012067  140212                  MOV     (R0)+,$TMP1     ;RETURN ON ERROR TO PRINT DATA
10122  040766  011000                          MOV     (R0),R0         ;RETURN ON NO ERROR
10123  040770  010304                          MOV     R3,R4           ;NO OF WORDS TO BE COMPARED
10124  040772  005204                          INC     R4
10125  040774  010437  044532          1$:     MOV     R4,@#ERWORD     ;FOR ERROR WORD NO
10126  041000  022122                          CMP     (R1)+,(R2)+     ;COMPARE GOOD WITH TEST DATA
10127  041002  001426                          BEQ     3$              ;BRANCH IF GOOD
10128
10129  041004  014137  001124                  MOV     -(R1),@#$GDDAT  ;GOOD DATA
10130  041010  014237  001126                  MOV     -(R2),@#$BDDAT  ;BAD DATA
10131  041014  160337  044532                  SUB     R3,@#ERWORD     ;ERROR WORD NO.
10132  041020  005737  001772                  TST     @#ERFLG$        ;ANY ERRORS ALREAY THERE
10133  041024  001003                          BNE     2$              ;BRANCH IF YES
10134  041026  004777  140144                  JSR     PC,@$TMP0       ;RETURN TO PRINT HEADER
10135  041032  000402                          BR      5$              ;BRANCH TO AVOID PRINTING NEXT ERROR
10136  041034  004777  140140          2$:     JSR     PC,@$TMP1       ;RETURN TO PRINT DATA
10137  041040  022122                  5$:     CMP     (R1)+,(R2)+     ;UNDO -(R1) AND -(R2) FOR ERRORS
10138  041042  013746  001140                  MOV     @#SWR,-(SP)     ;GET SWITCH SETTING
10139  041046  042716  177177                  BIC     #^C600,(SP)     ;KEEP ONLY SWITCH 7 AND 8
10140  041052  022726  000200                  CMP     #SW07,(SP)+     ;IS 7 SET AND 8 RESET
10141  041056  001402                          BEQ     4$              ;BRANCH OUT IF YES
10142  041060  005303                  3$:     DEC     R3              ;COUNT
10143  041062  001344                          BNE     1$              ;BRANCH IF ALL NOT DEVICE
10144  041064                          4$:
10145  041064  012605                          MOV     (SP)+,R5        ;;POP STACK INTO R5
10146  041066  012604                          MOV     (SP)+,R4        ;;POP STACK INTO R4
10147  041070  012603                          MOV     (SP)+,R3        ;;POP STACK INTO R3
10148  041072  012602                          MOV     (SP)+,R2        ;;POP STACK INTO R2
10149  041074  012601                          MOV     (SP)+,R1        ;;POP STACK INTO R1
10150  041076  000200                          RTS     R0              ;RETURN TO MAIN PROGRAM
10151
10152
10153
10154
10155                                   ;THIS IS A SUBROUTINE TO DO WRITE CHECK DATA
10156                                   ;CYLINDER 0, TRACK 1, SECTOR 1, KEYS 0
10157
10158                                   ;THESE ARE TO SET UP FOR DISKLESS USE ONLY
10159  041100  012737  010000  044412  WRCHDA: MOV     #FMT22,@#CYL    ;CYLINDER 0 FORMAT 16 BIT WORDS
10160  041106  112737  000001  044415          MOVB    #1,@#SECOTR+1   ;TRACK=1
10161  041114  112737  000001  044414          MOVB    #1,@#SECOTR     ;SECTOR=1
10162  041122  005037  044416                  CLR     @#KEY1          ;KEY1=0
10163  041126  005037  044420                  CLR     @#KEY2          ;KEY2=0
```

```
10164  041132  012737  000040  044472          MOV     #32.,@#DAWORD   ;NO OF DATA WORDS
10165  041140  005037  044422                  CLR     @#X             ;THIS IS A READ OPERATION
10166
10167  041144  004537  041242                  JSR     R5,@#CRC        ;GO TO CALCULATE CRC
10168  041150  044412                          CYL
10169  041152  046312                          WCRC
10170
10171                                          ;THESE ARE REGULAR SETUPS
10172
10173  041154  004737  040064                  JSR     PC,@#CLDISK     ;SET UP GENERAL REGISTERS
10174                                                                  ;AND CLEAR DISK REGISTERS
10175
10176  041160  012777  177740  140434          MOV     #-32.,@RHWC     ;36 DATA WORDS 4 HEADER WORDS
10177  041166  012777  003126  140430          MOV     #REINTO,@RHBA   ;STARTING ADDRESS OF READ BUFFER
10178  041174  112746  000001                  MOVB    #1,-(SP)        ;SECTOR=1
10179  041200  112766  000001  000001          MOVB    #1,1(SP)        ;TRACK=1 IN UPPER BYTE
10180  041206  012677  140422                  MOV     (SP)+,@RHDST    ;TRACK=1, SECTOR=1 IN RHDST
10181  041212  012777  014000  140420          MOV     #FMT22!ECI,@RHOF        ;16 BIT WORDS
10182                                                                  ;ECC CORRECTION INHIBIT BECAUSE
10183                                                                  ;ECC LOGIC IS NOT CHECKED YET
10184  041220  005077  140416                  CLR     @RHCA           ;CYLINDER=0
10185  041224  004737  040120                  JSR     PC,@#CHECKT     ;CHECK FOR DVA,RDY,MOL,DPR,DRY
10186  041230  013711  002032                  MOV     @#WRCHEK,@R1    ;WRITE CHECK DATA=50 INTO RHCS1
10187  041234  004737  044302                  JSR     PC,@#COMHD      ;WRITE CHECK HEADER AND DATA
10188                                                                  ;SAME AS READ HEADER AND DATA
10189
10190  041240  000207                          RTS     PC              ;RETURN TO WRITE CHECK TEST
10191
10192                                          .SBTTL  CRC GENERATION ROUTINE
10193                                          ;THIS IS A SUBROUTINE TO CALCULATE CRC FOR THE FOUR
10194                                          ;HEADER WORDS AND STORE THEM IN "WCRC" AND "GCRC"
10195                                          ;R1 - REGISTER FOR CRC, INCREMENTED CRC VALUE IS HERE
10196                                          ;R2 - THIS HAS BIT POSITION 2 VALUE C
10197                                          ;R3 - THIS HAS BIT POSITION 16 I.E. OUTPUT BIT VALUE B
10198                                          ;R4 - THIS HAS BIT POSITION 15 VALUE E
10199                                          ;$TMP0 - NUMBER OF WORDS
10200                                          ;$TMP2 - NUMBER OF BITS PER WORD = 16
10201                                          ;$TMP3 - TEMPORARY REG.
10202                                          ;$TMP4 - TEMPORARY REG TO TRANSFER CARRY
10203                                          ;$TMP5 - THIS HAS DATA BIT VALUE D
10204
10205                                          ;FETCH DATA BIT D
10206                                          ;B = D XOR 16
10207                                          ;C = B XOR 2
10208                                          ;E = B XOR 15
10209                                          ;ROTATE RIGHT ONE POSITION
10210                                          ;B GOES TO POSITION 1
10211                                          ;C GOES TO POSITION 3
10212                                          ;E GOES TO POSITION 16
10213                                          ;REPET 64 TIMES
10214                                          ;CALL   JSR     R5,@#CRC
10215                                          ;X      ;FIRST LOCATION AT
10216                                          ;Y      ;PUT CRC IN WCRC FOR READ GCRC FOR WRITE
10217
10218  041242                          CRC:
10219  041242  010046                          MOV     R0,-(SP)        ;;PUSH R0 ON STACK
```

```
10220  041244  012500                        MOV     (R5)+,R0        ;GET POINTER TO CYL NO.
10221  041246  010146                        MOV     R1,-(SP)        ;;PUSH R1 ON STACK
10222  041250  010246                        MOV     R2,-(SP)        ;;PUSH R2 ON STACK
10223  041252  010346                        MOV     R3,-(SP)        ;;PUSH R3 ON STACK
10224  041254  010446                        MOV     R4,-(SP)        ;;PUSH R4 ON STACK
10225  041256  005001                        CLR     R1              ;CLEAR WORKING LOCATION
10226  041260  005037  001210                CLR     @#$TMP5
10227  041264  012737  000004  001176        MOV     #4,@#$TMP0      ;WORD COUNT
10228  041272  012037  001204        16$:    MOV     (R0)+,@#$TMP3   ;TEMPORARY WORD STORAGE
10229  041276  012767  000020  137676        MOV     #16.,$TMP2      ;BIT COUNT
10230  041304  013737  001204  001206        MOV     @#$TMP3,@#$TMP4 ;TEMPORARY WORD STORAGE
10231  041312  006037  001204        15$:    ROR     @#$TMP3         ;GET LSB INTO "C"
10232  041316  006037  001210                ROR     @#$TMP5         ;GET ABOVE "C" INTO $TMP5
10233  041322  032701  000001                BIT     #BIT0,R1        ;IS POSITION 15 HIGH
10234  041326  001403                        BEQ     1$              ;BRANCH IF POSITION 16 LOW
10235  041330  012703  100000                MOV     #BIT15,R3       ;GET POSITION 16
10236  041334  000401                        BR      2$
10237  041336  005003                1$:     CLR     R3              ;GET POSITION 16
10238  041340  063703  001210        2$:     ADD     @#$TMP5,R3      ;XOR POSITION 16 WITH D
10239                                                                ;TO GIVE B
10240  041344  032701  040000                BIT     #BIT14,R1       ;IS POSITION 2 HIGH
10241  041350  001403                        BEQ     3$              ;BRANCH IF POSITION 2 LOW
10242  041352  012702  100000                MOV     #BIT15,R2       ;GET POSITION 2
10243  041356  000401                        BR      4$
10244  041360  005002                3$:     CLR     R2              ;GET POSITION 2
10245  041362  060302                4$:     ADD     R3,R2           ;XOR B WITH POSITION 2
10246                                                                ;TO GIVE C
10247  041364  032701  000002                BIT     #BIT1,R1        ;IS POSITION 15 HIGH
10248  041370  001403                        BEQ     5$              ;BRANCH IF POSITION 15 LOW
10249  041372  012704  100000                MOV     #BIT15,R4       ;GET POSITION 15
10250  041376  000401                        BR      6$
10251  041400  005004                5$:     CLR     R4              ;GET POSITION 15
10252  041402  060304                6$:     ADD     R3,R4           ;XOR POSITION 15 WITH B
10253                                                                ;TO GIVE E
10254  041404  006037  001206                ROR     @#$TMP4         ;GET LSB INTO "C"
10255  041410  006001                        ROR     R1              ;GET ABOVE C INTO R1
10256  041412  005703                        TST     R3              ;TEST B
10257  041414  100403                        BMI     7$              ;BRANCH IF B=1
10258  041416  042701  100000                BIC     #BIT15,R1       ;SET B IN POSITION 1
10259  041422  000402                        BR      10$
10260  041424  052701  100000        7$:     BIS     #BIT15,R1       ;SET B IN POSITION 1
10261  041430  005702                10$:    TST     R2              ;TEST C
10262  041432  100403                        BMI     11$             ;BRANCH IF C=1
10263  041434  042701  020000                BIC     #BIT13,R1       ;GET C IN POSITION 3
10264  041440  000402                        BR      12$
10265  041442  052701  020000        11$:    BIS     #BIT13,R1       ;GET C IN POSITION 3
10266  041446  005704                12$:    TST     R4              ;TEST E
10267  041450  100403                        BMI     13$             ;BRANCH IF E=1
10268  041452  042701  000001                BIC     #BIT0,R1        ;GET E IN POSITION 16
10269  041456  000402                        BR      14$
10270  041460  052701  000001        13$:    BIS     #BIT0,R1        ;GET E IN POSITION 16
10271  041464  005337  001202        14$:    DEC     @#$TMP2         ;BIT COUNTER
10272  041470  001310                        BNE     15$             ;BRANCH IF 16 NOT DONE
10273  041472  005337  001176                DEC     @#$TMP0         ;WORD COUNTER
10274  041476  001275                        BNE     16$             ;BRANCH IF 4 NOT DONE
10275  041500  010135                        MOV     R1,@(R5)+       ;PUT CRC WHERE DESIRED
```

```
10276  041502  012604                          MOV     (SP)+,R4        ;;POP STACK INTO R4
10277  041504  012603                          MOV     (SP)+,R3        ;;POP STACK INTO R3
10278  041506  012602                          MOV     (SP)+,R2        ;;POP STACK INTO R2
10279  041510  012601                          MOV     (SP)+,R1        ;;POP STACK INTO R1
10280  041512  012600                          MOV     (SP)+,R0        ;;POP STACK INTO R0
10281  041514  000205                          RTS     R5
10282
10283
10284
10285
10286                                           ;THIS IS A SUBROUTINE TO SET UP THE SIMULATOR DISK FOR
10287                                           ;CYLINDER 0 (16 BITS PER WORD)
10288                                           ;TRACK 1, SECTOR 1
10289                                           ;KEY1 1
10290                                           ;KEY2 1
10291                                           ;CRC THROUGH THE JSR R5,@#CRC
10292                                           ;256 WORDS OF 177400
10293
10294                                  ;CALL    JSR     PC,@#SETDSK
10295
10296  041516                          SETDSK:
10297  041516  010046                          MOV     R0,-(SP)        ;;PUSH R0 ON STACK
10298  041520  010146                          MOV     R1,-(SP)        ;;PUSH R1 ON STACK
10299  041522  010246                          MOV     R2,-(SP)        ;;PUSH R2 ON STACK
10300  041524  012700  177400                  MOV     #177400,R0      ;DATA IN THE DISK
10301  041530  012701  000400                  MOV     #256.,R1        ;COUNTER
10302  041534  012702  046330                  MOV     #DISK,R2        ;START OF SIMULATOR DISK
10303  041540  010022                  1$:     MOV     R0,(R2)+        ;MOVE IN DATA
10304  041542  005301                          DEC     R1              ;COUNT FOR 256
10305  041544  001375                          BNE     1$              ;BRANCH IF 256 NOT COMPLETE
10306  041546  012701  000021                  MOV     #17.,R1         ;2 ECC WORDS, 1 DATA GAP
10307                                                                  ;14 TOLERANCE GAP
10308  041552  005022                  2$:     CLR     (R2)+           ;CLEAR ECC,DATA GAP AND
10309                                                                  ;TOLERANCE GAP
10310  041554  005301                          DEC     R1              ;COUNT
10311  041556  001375                          BNE     2$              ;BRANCH IF NOT COMPLETE
10312
10313                                           ;NOW SET UP FOR DISKLESS USE
10314
10315  041560  012737  010000  044412          MOV     #FMT22,@#CYL    ;CYLINDER 0 (16 BIT WORDS)
10316  041566  112737  000001  044415          MOVB    #1,@#SECOTR+1   ;TRACK=1
10317  041574  112737  000001  044414          MOVB    #1,@#SECOTR     ;SECTOR=1
10318  041602  012737  000001  044416          MOV     #1,@#KEY1       ;KEY1=1
10319  041610  012737  000001  044420          MOV     #1,@#KEY2       ;KEY2=1
10320  041616  016737  136556  044472          MOV     256.,@#DAWORD   ;NO. OF DATA WORDS
10321  041624  004537  041242                  JSR     R5,@#CRC        ;GO TO CALCULATE CRC
10322  041630  044412                          CYL                     ;FIRST CRC WORD
10323  041632  046312                          WCRC                    ;PUT CALCULATED CRC
10324  041634  012602                          MOV     (SP)+,R2        ;;POP STACK INTO R2
10325  041636  012601                          MOV     (SP)+,R1        ;;POP STACK INTO R1
10326  041640  012600                          MOV     (SP)+,R0        ;;POP STACK INTO R0
10327  041642  000207                          RTS     PC
10328
10329                                           ;THIS IS A SUBROUTINE TO CHECK HEADER COMPARE ERROR
10330                                           ;(BIT #7) AND CRC ERROR (BIT #8)
10331                                           ;CALL JSR R0,@#HCCRCE
```

```
10332                                          ;       COM             ;COMMAND-READ HEADER AND DATA
10333                                          ;                       ;       -WRITE DATA
10334                                          ;       C               ;CYLINDER
10335                                          ;       S               ;SECTOR
10336                                          ;       T               ;TRACK
10337                                          ;       -N.             ;WORD COUNT
10338                                          ;       B               ;RHBA BUFFER START
10339                                          ;       X               ;1=WRITE DATA 0=READ
10340                                          ;       H               ;H=1 HEADER CHECK, H=0 CRC CHECK
10341
10342  041644  010037  002000          HCCRCE: MOV     R0,@#PCJSR      ;SAVE PC OF JSR+4
10343  041650  162737  000004  002000          SUB     #4,@#PCJSR      ;GET PC OF JSR
10344  041656  004737  040064                  JSR     PC,@#CLDISK     ;INIT AND SETUP GENERAL REG.
10345
10346
10347  041662  004767  176232                  JSR     PC,CHECKT               ;CHECK DVA, RDY, DPR, DRY
10348
10349  041666  011037  001210                  MOV     (R0),@#$TMP5    ;SAVE COMMAND
10350  041672  012011                          MOV     (R0)+,@R1       ;COMMAND
10351  041674  012077  137742                  MOV     (R0)+,@RHCA     ;CYLINDER
10352  041700  112046                          MOVB    (R0)+,-(SP)     ;SECTOR
10353  041702  105720                          TSTB    (R0)+           ;UP DATE R0
10354  041704  112066  000001                  MOVB    (R0)+,1(SP)     ;TRACK
10355  041710  105720                          TSTB    (R0)+           ;UPDATE R0
10356  041712  012677  137716                  MOV     (SP)+,@RHDST    ;TRACK SECTOR
10357  041716  012077  137700                  MOV     (R0)+,@RHWC     ;NO. OF DATA WORDS +4 HEADER
10358                                                                  ;IF A READ HEADER AND DATA
10359  041722  012077  137676                  MOV     (R0)+,@RHBA     ;STARTING ADDRESS OF BUFFER
10360  041726  012037  044422                  MOV     (R0)+,@#X       ;X=0 READ HEADER AND DATA
10361                                                                  ;X=1 WRITE DATA
10362  041732  012777  014000  137700          MOV     #FMT22!ECI,@RHOF ;16 BITS PER WORD
10363                                                                  ;ECC CORRECTION INHIBIT
10364  041740  005037  001772                  CLR     @#ERFLG$        ;CLEAR ERROR FLAG
10365  041744  004737  044302                  JSR     PC,@#COMHD      ;COMMAND
10366
10367                                          ;IF THE PROGRAM COMES BACK HERE WITHOUT ERROR PRINTOUTS
10368                                          ;FROM THE "COMHD" ROUTINE THAT MEANS SECTOR GAP,
10369                                          ;FIRST SYNC, HEADER, HEADER CRC, HEADER GAP AND
10370                                          ;SYNC BYTE HAVE GONE BY AND SYNCS WERE CORRECTLY
10371                                          ;DETECTED
10372                                          ;HEADER AND DATA ARE TO BE CHECKED.
10373  041750  004737  037470                  JSR     PC,@#PUTREG     ;SAVE REGISTERS
10374  041754  005737  001772                  TST     @#ERFLG$        ;ANY ERRORS ALREADY THERE
10375  041760  001034                          BNE     10$             ;BRANCH IF YES
10376  041762  005737  044422                  TST     @#X             ;IS THIS A READ
10377  041766  001015                          BNE     3$              ;IF A WRITE DATA BRANCH
10378
10379                                          ;NOW THE READ BUFFER WILL BE CHECKED
10380                                          ;HEADER SHOULD BE COMPLETELY READ AS WRITTEN
10381                                          ;NO DATA WORDS SHOULD BE READ
10382                                          ;REINTO BUFFER HAS BEEN FILLED WITH 0
10383                                          ;WRFROM BUFFER HAS BEEN FILLED WITH EXPECTED DATA
10384
10385  041770  004037  040736                  JSR     R0,@#COMPAR     ;CHECK
10386  041774  002062                          WRFROM                  ;GOOD DATA
10387  041776  003126                          REINTO                  ;TEST BUFFER
```

```
10388  042000  000400                          256.                    ;4 HEADER 252 DATA
10389  042002  042010                          1$                      ;RETURN POINT FOR ERROR HEADER
10390  042004  042014                          2$                      ;RETURN POINT FOR ERROR DATA
10391  042006  042052                          10$                     ;RETURN FOR GOOD COMPARISON
10392  042010  104004                  1$:     ERROR   4               ;READ NEXT ERROR 5
10393  042012  000207                          RTS     PC              ;RETURN TO COMPARISON SUBRCUTINE
10394  042014  104005                  2$:     ERROR   5               ;WORD NO 1 THRU 4 ARE
10395                                                                  ;HEADER WORDS AND HENCE
10396                                                                  ;SHOULD BE READ AS WRITTEN ON
10397                                                                  ;DISK, WORD NOS. 5 ONWARDS
10398                                                                  ;SHOULD NOT BE READ AND HENCE
10399                                                                  ;READ INTO BUFFER
10400                                                                  ;SHOULD BE UNCHANGED
10401  042016  000207                          RTS     PC              ;RETURN TO COMPARISON
10402
10403  042020  000414                          BR      10$             ;JUMP OUT
10404
10405                                          ;NOW THE DISK WILL BE CHECKED
10406                                          ;NO DATA SHOULD BE WRITTEN
10407                                          ;REINTO BUFFER HAS BEEN FILLED WITH EXPECTED DATA
10408                                          ;DISK HAS BEEN FILLED WITH 177400
10409                                          ;WRFROM HAS BEEN FILLED WITH 125252
10410
10411  042022  004037  040736          3$:     JSR     R0,@#COMPAR     ;CHECK
10412  042026  003126                          REINTO                  ;GOOD DATA BUFFER
10413  042030  046330                          DISK                    ;TEST BUFFER
10414  042032  000400                          256.
10415  042034  042042                          4$                      ;RETURN POINT FOR ERROR HEADER
10416  042036  042046                          5$                      ;RETURN POINT FOR ERROR DATA
10417  042040  042052                          10$                     ;RETURN POINT FOR GOOD COMPARISON
10418  042042  104004                  4$:     ERROR   4               ;READ NEXT ERROR 5
10419  042044  000207                          RTS     PC              ;RETURN TO COMPARISON SUBROUTINE
10420  042046  104005                  5$:     ERROR   5               ;WORD NO ARE ALL DATA
10421                                                                  ;WORDS THE SHOULD NOT
10422                                                                  ;HAVE BEEN CHANGED BY THE
10423                                                                  ;WRITE COMMAND
10424  042050  000207                          RTS     PC              ;RETURN TO COMPARISON SUBROUTINE
10425  042052  005720                  10$:    TST     (R0)+           ;IS THIS A HCRC ON HCE CHECK?
10426  042054  001442                          BEQ     6$              ;BRANCH IF HCRC
10427  042056  022737  000072  001210          CMP     #72,@#$TMP5     ;IS THIS A READ COMMAND
10428  042064  001417                          BEQ     11$             ;BRANCH IF YES
10429  042066  017737  137540  001126          MOV     @RHER1,@#$BDDAT ;TEST DATA
10430  042074  022737  000200  001126          CMP     #HCE,@#$BDDAT   ;ONLY HEADER COMPARE BIT?
10431                                                                  ;SHOULD BE SET
10432  042102  001470                          BEQ     7$              ;BRANCH IF GOOD
10433  042104  013737  001632  037534          MOV     @#RHER1,@#REGADR ;REGISTER ADDRESS RHER1
10434  042112  012737  000200  001124          MOV     #HCE,@#$GDDAT   ;GOOD DATA
10435  042120  104027                          ERROR   27              ;AFTER AN ERROR ON THE
10436                                                                  ;HEADER ONLY HCE SHOULD
10437  042122  000460                          BR      7$              ;BE SET
10438  042124                          11$:
10439  042124  017737  137502  001126          MOV     @RHER1,@#$BDDAT ;TEST DATA
10440  042132  022737  100200  001126          CMP     #DCK!HCE,@#$BDDAT       ;ONLY HEADER COMPARE BIT?
10441                                                                  ;SHOULD BE SET
10442                                                                  ;DCK IS SET BECAUSE ECC IS NOT READ
10443  042140  001451                          BEQ     7$              ;BRANCH IF GOOD
```

```
10444  042142  013737  001632  037534          MOV     @#RHER1,@#REGADR ;REGISTER ADDRESS RHER1
10445  042150  012737  100200  001124          MOV     #DCK!HCE,@#$GDDAT       ;GOOD DATA
10446  042156  104027                          ERROR   27                      ;AFTER AN ERROR ON THE
10447                                                                          ;HEADER ONLY HCE SHOULD
10448  042160  000441                          BR      7$                      ;BE SET
10449  042162  022737  000072  001210  6$:     CMP     #72,@#$TMP5             ;IS THIS A READ COMMAND?
10450  042170  001417                          BEQ     12$                     ;BRANCH IF A READ
10451  042172  017737  137434  001126          MOV     @RHER1,@#$BDDAT ;TEST DATA
10452  042200  022737  000400  001126          CMP     #HCRC,@#$BDDAT  ;ONLY CRC ERROR SHOULD BE THERE
10453  042206  001426                          BEQ     7$
10454  042210  013737  001632  037534          MOV     @#RHER1,@#REGADR ;REG. ADDR = RHER1
10455  042216  012737  000400  001124          MOV     #HCRC,@#$GDDAT  ;GOOD DATA
10456  042224  104027                          ERROR   27                      ;AFTER A CRC ERROR ONLY CRC
10457                                                                          ;SHOULD BE SET
10458  042226  000416                          BR      7$                      ;BRANCH OUT
10459  042230  017737  137376  001126  12$:     MOV     @RHER1,@#$BDDAT ;TEST DATA
10460
10461  042236  022737  100400  001126          CMP     #DCK!HCRC,@#$BDDAT;HCRC AND DCK SHOULD BE SET
10462                                                                          ;DCK IS SET BECAUSE ECC IS NOT READ
10463  042244  001407                          BEQ     7$                      ;BRANCH IF GOOD
10464  042246  012737  100400  001124          MOV     #DCK!HCRC,@#$GDDAT;GOOD DATA
10465  042254  013737  001632  037534          MOV     @#RHER1,@#REGADR;FAILING REGISTER RHER1
10466  042262  104027                          ERROR   27                      ;AFTER A CRC ERROR ON A READ
10467                                                                          ;DCK AND HCRC SHOULD BE SET
10468                                                                          ;DCK IS SET BECAUSE ECC IS NOT READ
10469  042264  000200                  7$:     RTS     R0                      ;RETURN TO MAIN TEST
10470
10471
10472
10473                                          ;THIS IS A SUBROUTINE TO LEAVE AT THE MIDDLE OF
10474                                          ;A WRITE HEADER AND DATA COMMAND
10475                                          ;IT TRYS TO GET SECTOR 10, TRACK 0, CYLINDER 0
10476                                          ;BUT COMES OUT AFTER ONE SECTOR
10477                                          ;THE COMMAND OS JSR PC,@#MIDDLE
10478                                          ;BAI IS SET
10479
10480  042266                          MIDDLE:
10481  042266  010046                          MOV     R0,-(SP)                ;;PUSH R0 ON STACK
10482  042270  010146                          MOV     R1,-(SP)                ;;PUSH R1 ON STACK
10483  042272  013777  002040  137330          MOV     @#WRIFOR,@RHCS1 ;WRITE HEADER AND DATA=62
10484                                                                          ;IN RHCS1
10485  042300  012777  177766  137314          MOV     #-10.,@RHWC             ;10 WORDS
10486  042306  012777  002062  137310          MOV     #WRFROM,@RHBA           ;BUS ADDRESS=WRFROM
10487  042314  012777  000010  137312          MOV     #10,@RHDST              ;DESIRED TRACK=0 SECTOR=10
10488  042322  052777  000010  137276          BIS     #BAI,@RHCS2             ;BUS ADDRESS INCREMENT INHIBIT
10489  042330  012777  010000  137302          MOV     #FMT22,@RHOF            ;FORMAT 16 BIT WORDS
10490  042336  005077  137300                  CLR     @RHCA                   ;CYLINDER=0
10491  042342  012737  000001  042370          MOV     #1,@#MID                ;SECTOR IS SET TO 1 SO THAT
10492                                                                          ;WE CAN GET OUT AT THE
10493                                                                          ;MIDDLE OF AN OPERATION
10494                                                                          ;LOOKING FOR SECTOR 10
10495  042350  012777  000001  137272          MOV     #DMD,@RHMR              ;SET DIAGNOSTIC MODE
10496  042356  052777  000001  137244          BIS     #GO,@RHCS1                      ;GO TO RHCS1 WITH 62
10497  042364  004137  050426                  JSR     R1,@#SEARCH
10498  042370  000000                  MID:    .WORD   0                       ;SECTOR
10499  042372  012601                          MOV     (SP)+,R1                ;;POP STACK INTO R1
```

```
10500  042374  012600                      MOV     (SP)+,R0        ;;POP STACK INTO R0
10501  042376  000207                      RTS     PC
10502
10503
10504
10505                                          .SBTTL  JAM CURRENT CYLINDER ROUTINE
10506                                  ;THIS SUBROUTINE WILL CHANGE THE CURRENT CYLINDER REGISTER
10507                                  ;THIS IS DONE BY GIVING A SEEK COMMAND THEN AN INIT
10508                                  ;WHICH WILL LOAD THE CURRENT CYLINDER WITH THE DESIRED CYLINDER VALUE
10509                                  ;
10510                                  ;CALL IS
10511                                  ;JSR    R0,@#MAKECYL
10512                                  ;XC             ;DESIRED VALUE OF CURRENT CYLINDER
10513
10514
10515  042400                          MAKECYL:
10516  042400  010546                          MOV     R5,-(SP)        ;;PUSH R5 ON STACK
10517  042402  010037  002000                  MOV     R0,@#PCJSR      ;PC OF JSR+4
10518  042406  162737  000004  002000          SUB     #4,@#PCJSR      ;SAVE PC OF JSR
10519  042414  012005                          MOV     (R0)+,R5        ;GETTING READY TO FILL DESIRED CYLINDER
10520  042416  010577  137220                  MOV     R5,@RHCA        ;FILL DESIRED CYLINDER REGISTER
10521  042422  005077  137206                  CLR     @RHDST          ;MAKE SURE DESIRED SECTOR TRACK IS NOT ILLEGAL
10522  042426  013777  002046  137174          MOV     @#SEECOM,@RHCS1 ;FILL SEEK COMMAND
10523  042434  012777  000001  137206          MOV     #DMD,@RHMR      ;SET DIAGNOSTIC MODE
10524  042442  052777  000001  137160          BIS     #GO,@RHCS1      ;GO TO SEEK
10525  042450  000240                          NOP                     ;ALLOW TIME FOR SEEK TO HANG UP
10526  042452  000240                          NOP                     ;ALLOW TIME FOR SEEK TO HANG UP
10527  042454  000240                          NOP                     ;ALLOW TIME FOR SEEK TO HANG UP
10528  042456  000240                          NOP                     ;ALLOW TIME FOR SEEK TO HANG UP
10529  042460  004737  040064                  JSR     PC,@#CLDISK     ;GIVE INIT
10530  042464  017737  137176  001126          MOV     @RHCC,@#$BDDAT  ;TEST DATA
10531  042472  020537  001126                  CMP     R5,@#$BDDAT     ;COMPARE CURRENT CYLINDER
10532  042476  001406                          BEQ     1$              ;BRANCH IF GOOD
10533  042500  010537  001124                  MOV     R5,@#$GDDAT     ;GOOD VALUE OF RHCC
10534  042504  013737  001666  037534          MOV     @#RHCC,@#REGADR ;FAILING REGISTER ADDRESS
10535  042512  104030                          ERROR   30              ;CURRENT CYLINDER DOES NOT MATCH DESIRED CYLINDER
10536                                                                  ;REGISTER AFTER A SEEK AND AN INIT
10537  042514                          1$:
10538  042514  012605                          MOV     (SP)+,R5        ;;POP STACK INTO R5
10539  042516  000200                          RTS     R0
10540
10541
10542                                          .SBTTL  ECC GENERARION AND COMPARISON ROUTINE
10543
10544
10545
10546
10547                                  ;THIS SUBROUTINE GENERATES AND TESTS ECC
10548                                  ;CALL    JSR PC,ECTEST
10549
10550
10551
10552
10553
10554          100000                  PIE1    =100000
10555          040000                  PIE2    =40000
```

```
 10556           020000                  PIE3    =20000
 10557           010000                  PIE4    =10000
 10558           004000                  PIE5    =4000
 10559           002000                  PIE6    =2000
 10560           001000                  PIE7    =1000
 10561           000400                  PIE8    =400
 10562           000200                  PIE9    =200
 10563           000100                  PIE10   =100
 10564           000040                  PIE11   =40
 10565           000020                  PIE12   =20
 10566           000010                  PIE13   =10
 10567           000004                  PIE14   =4
 10568           000002                  PIE15   =2
 10569           000001                  PIE16   =1
 10570           100000                  PIE17   =100000
 10571           040000                  PIE18   =40000
 10572           020000                  PIE19   =20000
 10573           010000                  PIE20   =10000
 10574           004000                  PIE21   =4000
 10575           002000                  PIE22   =2000
 10576           001000                  PIE23   =1000
 10577           000400                  PIE24   =400
 10578           000200                  PIE25   =200
 10579           000100                  PIE26   =100
 10580           000040                  PIE27   =40
 10581           000020                  PIE28   =20
 10582           000010                  PIE29   =10
 10583           000004                  PIE30   =4
 10584           000002                  PIE31   =2
 10585           000001                  PIE32   =1
 10586
 10587  042520   000000                  ECDATA: 0               ;DATA BIT FOR ECC
 10588                                                            ;IF ALL ONES THEN CURRENT BIT IS A ONE
 10589                                                            ;IF ZERO THEN CURRENT BIT IS A ZERO
 10590
 10591  042522   000000                  GECC1:  0               ;LOW ORDER ECC WORD TO BE GENERATED HERE
 10592                                                            ;=R1
 10593
 10594  042524   000000                  GECC2:  0               ;HIGH ORDER ECC WORD TO BE GENERATED HERE
 10595                                                            ;=R2
 10596
 10597  042526   000000                  TSECCG: 0               ;IF =177777 GENERATE AND TEST ECC FOR THIS BIT
 10598                                                            ;IF =0 DO NOT GENERATE AND TEST ECC FOR THIS BIT
 10599
 10600  042530   113713                  NCODE:  38859.          ;N-CODE WORD
 10601  042532   000000                  NCOUNT: 0       ;TEMPORARY N CODE
 10602  042534   000000                  POSITI: 0               ;POSITION REGISTER
 10603  042536   010041                  HARDER: 4129.           ;HARD ERROR COUNT
 10604                                                            ;TRUE COUNT IS 4128 BUT AS COMPARES ARE
 10605                                                            ;DONE ONE STAGE LATER SO 4129
 10606  042540   000000                  DATENV: 0               ;DATA ENVELOPE FOR TYPE OUT
 10607                                                            ;MAX FOR WRITE IS 4096
 10608                                                            ;MAX FOR READ IS 4128
 10609  042542   000000                  ZCODE:  0               ;LEADING ZEROS ENVELOPE FOR TYPE OUT
 10610                                                            ;THIS IS SHUT OFF WHEN POSITION COUNTER
 10611                                                            ;IN ENABLED
```

```
10612                                                   ;MAX COUNT IS 38859
10613
10614
10615
10616  042544  000000                 HADTMP: 0       ;TEMPORARY HARD ERROR COUNT
10617  042546  000000                 P3:     0
10618  042550  000000                 P12:    0
10619  042552  000000                 P22:    0
10620  042554  000000                 P24:    0
10621
10622
10623
10624
10625
10626  042556                         ECTEST:
10627  042556  010046                         MOV     R0,-(SP)        ;;PUSH R0 ON STACK
10628  042560  010146                         MOV     R1,-(SP)        ;;PUSH R1 ON STACK
10629  042562  010246                         MOV     R2,-(SP)        ;;PUSH R2 ON STACK
10630  042564  010346                         MOV     R3,-(SP)        ;;PUSH R3 ON STACK
10631  042566  010446                         MOV     R4,-(SP)        ;;PUSH R4 ON STACK
10632  042570  010546                         MOV     R5,-(SP)        ;;PUSH R5 ON STACK
10633  042572  013701  042522                 MOV     @#GECC1,R1              ;ECC1 WORD
10634  042576  013702  042524                 MOV     @#GECC2,R2              ;ECC2 WORD
10635  042602  005737  042520                 TST     @#ECDATA                ;IS CURRENT BIT A ONE
10636  042606  001406                         BEQ     2$                      ;BRANCH IF CURRENT DATA D=0
10637                                         ;IF CARRY IS NOT ZERO THEN D=1
10638                                         ;INVERT X32 TO GIVE R0
10639  042610  010103                 1$:     MOV     R1,R3
10640  042612  052703  177776                 BIS     #^CPIE32,R3
10641  042616  005103                         COM     R3
10642  042620  010300                         MOV     R3,R0
10643  042622  000404                         BR      3$
10644
10645                                         ;IF CARRY IS ZERO THEN D=0
10646                                         ;X32 BECOMES R0
10647  042624  010103                 2$:     MOV     R1,R3
10648  042626  042703  177776                 BIC     #^CPIE32,R3
10649  042632  010300                         MOV     R3,R0
10650
10651  042634  000241                 3$:     CLC
10652  042636  006000                         ROR     R0
10653  042640  006000                         ROR     R0
10654  042642  005700                         TST     R0
10655  042644  001462                         BEQ     10$             ;BRANCH IF R0=0
10656                                         ;INVERT X2
10657
10658  042646  010203                         MOV     R2,R3
10659  042650  052703  137777                 BIS     #^CPIE2,R3
10660  042654  005103                         COM     R3
10661  042656  010337  042546                 MOV     R3,@#P3
10662  042662  006237  042546                 ASR     @#P3
10663
10664                                         ;INVERT X11
10665
10666
10667  042666  010203                         MOV     R2,R3
```

```
10668  042670  052703  177737          BIS     #↑CPIE11,R3
10669  042674  005103                  COM     R3
10670  042676  010337  042550          MOV     R3,@#P12
10671  042702  006237  042550          ASR     @#P12
10672
10673                                  ;INVERT X21
10674
10675  042706  010103                  MOV     R1,R3
10676  042710  052703  173777          BIS     #↑CPIE21,R3
10677  042714  005103                  COM     R3
10678  042716  010337  042552          MOV     R3,@#P22
10679  042722  006237  042552          ASR     @#P22
10680
10681                                  ;INVERT X23
10682
10683  042726  010103                  MOV     R1,R3
10684  042730  052703  176777          BIS     #↑CPIE23,R3
10685  042734  005103                  COM     R3
10686  042736  010337  042554          MOV     R3,@#P24
10687  042742  006237  042554          ASR     @#P24
10688
10689                                  ;NOW THAT R0 FOR POSITION 1
10690                                  ;         P3 FOR POSITION 3
10691                                  ;        P12 FOR POSITION 12
10692                                  ;        P22 FOR POSITION 22
10693                                  ;        P24 FOR POSITION 24
10694                                  ;ARE KNOWN THE ROTATE WILL BE DONE AND
10695                                  ;THESE BITS JAMED IN
10696
10697  042746  006002                  ROR     R2
10698  042750  006001                  ROR     R1
10699  042752  053700  042546          BIS     @#P3,R0
10700  042756  053700  042550          BIS     @#P12,R0
10701  042762  042702  120020          BIC     #PIE1!PIE3!PIE12,R2
10702  042766  050002                  BIS     R0,R2
10703
10704  042770  005000                  CLR     R0
10705  042772  053700  042552          BIS     @#P22,R0
10706  042776  053700  042554          BIS     @#P24,R0
10707  043002  042701  002400          BIC     #PIE22!PIE24,R1
10708  043006  050001                  BIS     R0,R1
10709  043010  000404                  BR      12$
10710
10711                                  ;THE PROGRAM COMES HERE IF R0=0
10712                                  ;SO AFTER ROTATE R0 GETS PUT INTO POSITION 1
10713  043012  006002          10$:    ROR     R2
10714  043014  006001                  ROR     R1
10715  043016  042702  100000          BIC     #PIE1,R2
10716  043022  010137  042522  12$:    MOV     R1,@#GECC1      ;SAVE ECC1
10717  043026  010237  042524          MOV     R2,@#GECC2      ;SAVE ECC2
10718  043032  005737  042526          TST     @#↑SECCG        ;IS HARDWARE TO BE CHECKED
10719                                                          ;IF =1777777 TEST HARDWARE
10720                                                          ;IF = 0 DO NOT TEST HARDWARE
10721  043036  001432                  BEQ     14$             ;BRANCH IF HARDWARE NOT TO BE CHECKED
10722
10723
```

```
10724                                          ;CHECK HARDWARE
10725  043040  032737  000400  001140          BIT     #SW8,@#SWR      ;IS SWITCH 8 SET
10726  043046  001005                          BNE     15$             ;BRANCH IF SW8 IS SET
10727  043050  032737  000100  001140          BIT     #SW6,@#SWR      ;IS SWITCH 6 SET
10728  043056  001401                          BEQ     15$             ;BRANCH IF SW6 IS NOT SET
10729  043060  000421                          BR      14$             ;IF SWITCH 8 IS NOT SET AND
10730                                                                  ;SWITCH 6 IS SET THEN
10731                                                                  ;DO NOT DO COMPARES
10732  043062  010146                  15$:    MOV     R1,-(SP)        ;GOOD PATTERN REGISTER
10733  043064  042716  174000                  BIC     #174000,(SP)    ;GET ONLY PATTERN BITS
10734  043070  022677  136566                  CMP     (SP)+,@RHEC2    ;COMPARE PATTERN REGISTER
10735  043074  001404                          BEQ     13$             ;BRANCH IF GOOD
10736                                          ;TO SAVE TIME
10737  043076  004737  037470                  JSR     PC,@#PUTREG     ;SAVE REGISTERS
10738  043102  104035                          ERROR   35              ;PATTERN REGISTER IN 11 BITS IN ERROR
10739  043104  000407                          BR      14$             ;BRANCH OUT
10740  043106  023777  042534  136544  13$:    CMP     @#POSITI,@RHEC1 ;COMPARE POISTION REGISTER
10741  043114  001403                          BEQ     14$             ;BRANCH IF GOOD
10742                                          ;TO SAVE TIME
10743  043116  004737  037470                  JSR     PC,@#PUTREG     ;SAVE REGISTERS
10744  043122  104035                          ERROR   35              ;POSITION REGISTER IN ERROR
10745                                                                  ;"DATA ENVLOP" GIVES NUMBER OF CLOCK
10746                                                                  ;PULSES FROM BEGINING OF COMMAND
10747                                                                  ;THAT IS THE CLOCKS IN THE R/W DATA FIELD ENVELOPE
10748                                                                  ;
10749                                                                  ;IN A WRITE THERE ARE 10000 OCTAL CLOCKS
10750                                                                  ;IN A READ THERE ARE 10040 OCTAL CLOCKS
10751                                                                  ;
10752                                                                  ;
10753                                                                  ;"N-CODE ZEROS" GIVE THE NUMBER OF CLOCKS
10754                                                                  ; GIVEN FOR THE LEADING ZEROS FIELD
10755                                                                  ;MAX COUNT IS 113713 OCTAL
10756                                                                  ;
10757                                                                  ;"GOOD POSITION" GIVES NUMBER OF CLOCKS
10758                                                                  ;GIVEN AFTER LEADING ZEROS WHICH IS FOR THE DATA
10759                                                                  ;FIELD
10760                                                                  ;MAX COUNT IS 10040 OR 10041 OCTAL
10761
10762
10763  043124                  14$:
10764  043124  012605                  ;MOV     (SP)+,R5        ;;POP STACK INTO R5
10765  043126  012604                   MOV     (SP)+,R4        ;;POP STACK INTO R4
10766  043130  012603                   MOV     (SP)+,R3        ;;POP STACK INTO R3
10767  043132  012602                   MOV     (SP)+,R2        ;;POP STACK INTO R2
10768  043134  012601                   MOV     (SP)+,R1        ;;POP STACK INTO R1
10769  043136  012600                   MOV     (SP)+,R0        ;;POP STACK INTO R0
10770  043140  000207                   RTS     PC
10771
10772
10773                          ;THIS SUBROUTINE WILL CONTROL THE ECC GENERATION ROUTINE
10774                          ;FOR ERROR CORRECTION PROCESS
10775                          ;CALL   JSR,    PC,@#ECORR
10776                          ;       XP      ;EXPECTED POSITION REGISTER WHEN CORRECTION IS COMPLETE
10777
10778
10779
```

```
10780  043142  000000                    ERPOS:  0               ;POSITION REG. WHEN CORRECTION IS COMPLETE
10781
10782
10783
10784  043144  010037  002000            ECORR:  MOV     R0,@#PCJSR      ;SAVE PC OF JSR + 4
10785  043150  162737  000004  002000            SUB     #4,@#PCJSR      ;SAVE PC OF JSR
10786  043156  012037  043142                    MOV     (R0)+,@#ERPOS   ;GET POSITION REG. WHEN CORRECTION IS COMPLETE
10787  043162  010146                            MOV     R1,-(SP)        ;;PUSH R1 ON STACK
10788  043164  013701  001650                    MOV     @#RHMR,R1       ;MAIMTENANCE REGISTER
10789  043170  012711  000001                    MOV     #DMD,@R1        ;SET DIAGNOSTIC MODE BIT
10790  043174  005037  042520                    CLR     @#ECDATA        ;ECC DATA IS ZERO
10791
10792
10793
10794  043200  005737  042534            1$:     TST     @#POSITI        ;IS  SOFTWARE POSITION NON ZERO
10795  043204  001007                            BNE     2$              ;BRANCH IF N-CODE S COMPLETE
10796  043206  005337  042532                    DEC     @#NCOUNT        ;DECREMENT N-CODE
10797  043212  001001                            BNE     6$              ;BRANCH IF N-CODE IS NOT COMPLETE
10798  043214  000403                            BR      2$              ;BRANCH  AS N-CODE IS COMPLETE
10799  043216  005237  042542            6$:     INC     @#ZCODE         ;INCREMENT CLOCKS GIVEN FOR LEADING ZEROS
10800  043222  000420                            BR      3$              ;BRANCH AS N-CODE IS NOT COMPLETE
10801                                                                    ;GO TO GIVE CLOCK AND TEST ECC
10802  043224  005237  042534            2$:     INC     @#POSITI        ;INCREMENT SOFTWARE POSITION
10803  043230  023737  043142  042534            CMP     @#ERPOS,@#POSITI;HAVE ENOUGH CLOCKS BEEN GIVEN TO DETECT ERROR
10804  043236  103012                            BHIS    3$              ;BRANCH IF MORE CLOCKS TO BE GIVEN
10805  043240  023737  042544  042534            CMP     @#HADTMP,@#POSITI;HAVE ENOUGH CLOCKS BEEN GIVEN FOR HARD ERROR
10806                                                                    ;THAT IS HAVE 4128 MORE CLOCKS BEEN GIVEN
10807  043246  001415                            BEQ     5$              ;BRANCH IF YES
10808  043250  032711  000400                    BIT     #ZER,@R1        ;CHECK ZERO DETECT BIT IN RHMR
10809  043254  001016                            BNE     4$              ;BRANCH IS ZER SET
10810                                             ;TO SAVE TIME
10811  043256  004737  037470                    JSR     PC,@#PUTREG     ;SAVE REGISTERS
10812  043262  104034                            ERROR   34              ;ZERO DETECT BIT NOT HIGH
10813                                                                    ;WHEN 21 BITS IN ECC 32 BIT REGISTER IS  0
10814
10815
10816  043264  052711  000002            3$:     BIS     #MCLK,@R1       ;SET CLOCK
10817  043270  042711  000002                    BIC     #MCLK,@R1       ;CLEAR CLOCK
10818  043274  004737  042556                    JSR     PC,@#ECTEST     ;GO TO GENERATE AND TEST ECC
10819  043300  000737                            BR      1$              ;CONTINUE
10820
10821                                            ;THIS EXTRA CLOCK IS TO BRING ECH HIGH
10822                                            ;AFTER THIS CLOCK POSITION REGISTER MAY BE 10040 OR 10041 OCTAL
10823  043302  052711  000002            5$:     BIS     #MCLK,@R1       ;SET CLOCK
10824  043306  042711  000002                    BIC     #MCLK,@R1       ;CLEAR CLOCK
10825
10826  043312                             4$:
10827  043312  012601                            MOV     (SP)+,R1        ;;POP STACK INTO R1
10828  043314  000200                            RTS     R0
10829
10830
10831
10832
10833
10834
10835                                    ;THIS SUBROUTINE GENERATES THE ECC FOR WHAT IS ON DISK AND INSERTS THEM
```

```
                                        ;ON LOCATIONS "DISK+1000" AND "DISK+1002"


043316                          FILLEC:
043316  010046                          MOV     R0,-(SP)        ;;PUSH R0 ON STACK
043320  010146                          MOV     R1,-(SP)        ;;PUSH R1 ON STACK
043322  010246                          MOV     R2,-(SP)        ;;PUSH R2 ON STACK
043324  010346                          MOV     R3,-(SP)        ;;PUSH R3 ON STACK
043326  010446                          MOV     R4,-(SP)        ;;PUSH R4 ON STACK
043330  010546                          MOV     R5,-(SP)        ;;PUSH R5 ON STACK
043332  005037  042534                  CLR     @#POSITI        ;CLEAR POSITION
043336  005037  042522                  CLR     @#GECC1         ;CLEAR GECC1
043342  005037  042524                  CLR     @#GECC2         ;CLEAR
043346  012701  046330                  MOV     #DISK,R1        ;POINTER TO DATA FOR ECC GENERATION
043352  012702  000400                  MOV     #256.,R2        ;COUNTER FOR NUMBER OF DATA WORDS
043356  012703  000020          9$:     MOV     #16.,R3         ;COUNTER FOR NUMBER OF BITS PER WORD
043362  012104                  10$:    MOV     (R1)+,R4        ;DATA IN R4
043364  006004                          ROR     R4              ;GET ONE DATA BIT IN CARRY
043366  103004                          BCC     11$             ;BRANCH IF DATA BIT IS ZERO
043370  012737  177777  042520          MOV     #-1,@#ECDATA    ;ECC DATA BIT IS A ONE
043376  000402                          BR      12$             ;BRANCH TO GENERATE ECC
043400  005037  042520          11$:    CLR     @#ECDATA        ;ECC DATA BIT IS A ZERO
043404  004737  042556          12$:    JSR     PC,@#ECTEST     ;GO TO GENERATE ECC
043410  005303                          DEC     R3              ;DECREMENT BIT COUNT
043412  001364                          BNE     10$             ;BRANCH IF 16 BITS NOT DONE
043414  005302                          DEC     R2              ;DECREMENT WORD COUNT
043416  001357                          BNE     9$              ;BRANCH IF 256 WORDS NOT DONE
043420  013737  042522  047330          MOV     @#GECC1,@#DISK+<256.*2>;INSERT ECC1 ON DISK
043426  013737  042524  047332          MOV     @#GECC2,@#DISK+<257.*2>;INSERT ECC2 ON DISK
043434  012605                          MOV     (SP)+,R5        ;;POP STACK INTO R5
043436  012604                          MOV     (SP)+,R4        ;;POP STACK INTO R4
043440  012603                          MOV     (SP)+,R3        ;;POP STACK INTO R3
043442  012602                          MOV     (SP)+,R2        ;;POP STACK INTO R2
043444  012601                          MOV     (SP)+,R1        ;;POP STACK INTO R1
043446  012600                          MOV     (SP)+,R0        ;;POP STACK INTO R0
043450  000207                          RTS     PC
```

```
                                        .SBTTL  RH BASE ADDRESS CHANGE ROUTINE
                                ;*      THIS ROUTINE WILL ALLOW THE CHANGE OF THE BASE
                                ;*      ADDRESS FROM 176700 TO ANY TYPED VALUE

043452                          BASECH:
043452  104400  043460                  TYPE    ,65$            ;;TYPE ASCIZ STRING
043456  000424                          BR      64$             ;;GET OVER THE ASCIZ
                                ;;65$:  .ASCIZ  <15><12>/PRESENT BASE ADDRESS OF REGISTERS IS/
043530                          64$:
043530  013746  001630                  MOV     @#RHCS1,-(SP)   ;GET READY TO TYPE OLD BASE
043534  104401                          TYPOC
043536  104400  043544                  TYPE    ,67$            ;;TYPE ASCIZ STRING
043542  000425                          BR      66$             ;;GET OVER THE ASCIZ
                                ;;67$:  .ASCIZ  <15><12>/TYPE NEW BASE ADDRESS FOLLOWED BY 'CR'/
```

```
10892  043616                          66$:
10893  043616  104407                      RDOCT
10894  043620  012700  001620              MOV     #RHDB,R0        ;GET STARTING ADDRESS OF RGISTERS
10895  043624  012701  000024              MOV     #20.,R1         ;NUMBER OF REGISTERS
10896  043630  042710  177700      1$:     BIC     #^C77,(R0)      ;CLEAR OLD BASE
10897  043634  051620                      BIS     (SP),(R0)+      ;SET NEW BASE
10898  043636  005301                      DEC     R1              ;COUNT
10899  043640  001373                      BNE     1$              ;BRANCH IF 20 NOT DONE
10900  043642  104400  043650              TYPE    ,69$            ;;TYPE ASCIZ STRING
10901  043646  000417                      BR      68$             ;;GET OVER THE ASCIZ
10902                             ;;69$:   .ASCIZ  <15><12>/PRESENT VECTOR ADDRESS IS /
10903  043706                          68$:
10904  043706  013746  001616              MOV     @#RPVEC,-(SP)   ;GET READY TO TYPE OLD VECTOR ADDRESS
10905  043712  104401                      TYPOC
10906  043714  104400  043722              TYPE    ,71$            ;;TYPE ASCIZ STRING
10907  043720  000437                      BR      70$             ;;GET OVER THE ASCIZ
10908                             ;;71$:   .ASCIZ  <15><12>/TYPE NEW VECTOR ADDRESS OR RETYPE OLD ONE FOLLOWED BY "CR"/
10909  044020                          70$:
10910  044020  104407                      RDOCT
10911  044022  012637  001616              MOV     (SP)+,@#RPVEC   ;SETUP VECTOR ADDRESS
10912  044026  104400  044034              TYPE    ,73$            ;;TYPE ASCIZ STRING
10913  044032  000421                      BR      72$             ;;GET OVER THE ASCIZ
10914                             ;;73$:   .ASCIZ  <15><12>/RESTART PROGRAM FROM 200 OR 210/
10915  044076                          72$:
10916  044076  104400  044104              TYPE    ,75$            ;;TYPE ASCIZ STRING
10917  044102  000414                      BR      74$             ;;GET OVER THE ASCIZ
10918                             ;;75$:   .ASCIZ  <15><12>/NEW BASE WILL REMAIN/
10919  044134                          74$:
10920  044134  013746  001630              MOV     @#RHCS1,-(SP)
10921  044140  104401                      TYPOC
10922  044142  104400  044150              TYPE    ,77$            ;;TYPE ASCIZ STRING
10923  044146  000415                      BR      76$             ;;GET OVER THE ASCIZ
10924                             ;;77$:   .ASCIZ  <15><12>/NEW VECTOR WILL REMAIN /
10925  044202                          76$:
10926  044202  013746  001616              MOV     @#RPVEC,-(SP)
10927  044206  104401                      TYPOC
10928  044210  104400  044216              TYPE    ,79$            ;;TYPE ASCIZ STRING
10929  044214  000416                      BR      78$             ;;GET OVER THE ASCIZ
10930                             ;;79$:   .ASCIZ  <15><12>/UNTIL PROGRAM IS RELOADED/
10931  044252                          78$:
10932  044252  000000                      HALT
10933
10934
10935
10936
10937
10938                           ;THIS IS A LITTLE ROUTINE THAT TESTS NED BIT 11 IN RHCS2
10939                           ;THIS LOOPS HERE FOR EVER
10940                           ;TO BE USED ONLY IF DRIVES PRESENT LOOKING AT NED DOES NOT AGREE
10941                           ;WITH WHAT IS REALY THERE
10942  044254  000000          ERUNIT: 0                       ;UNIT UNDER MANUAL TEST
10943  044256  004737  040064  ERSTART:JSR     PC,@#CLDISK     ;SET GENERAL REG.
10944  044262  013712  044254          MOV     @#ERUNIT,@R2    ;SELECT UNIT
10945  044266  005714          1$:     TST     @R4             ;TEST RHER1
10946  044270  032712  010000          BIT     #NED,@R2        ;TEST NED
10947  044274  001401                  BEQ     2$              ;BRANCH IF GOOD
```

F01


```
10948  044276  000773                        BR      1$              ;NED NOT SET
10949  044300  000772                2$:     BR      1$              ;NED SET
```

```
10950                                   .SBTTL  DISK SIMULATION
10951                                   ;;*****************************************************************
10952                                   ;;*****************************************************************
10953                                   ;IN A WRITE HEADER AND DATA COMMAND FILL THE FOLLOWING
10954                                   ;WCLY=WITH CYLINDER TO BE ON DISK
10955                                   ;WSECTR=WITH SECTOR AND TRACK TO BE ON DISK
10956                                   ;WKEY1= WITH KEY1 TO BE ON DISK
10957                                   ;WKEY2= WITH KEY2 TO BE ON DISK
10958                                   ;FNWORD= NO OF DATA WORDS TO BE WRITTEN ON DISK
10959                                   ;THE COMMAND THEN IS   JSR PC,COMWHD
10960                                   ;
10961                                   ;
10962                                   ;
10963                                   ;
10964                                   ;IN A WRITE DATA COMMAND FILL THE FOLLOWING
10965                                   ;CYL=WITH CYLINDER TO BE FOUND ON DISK
10966                                   ;SECOTR= WITH SECTOR AND TRACK TO BE FOUND ON DISK
10967                                   ;KEY1= WITH KEY1 TO BE FOUND ON DISK
10968                                   ;KEY2= WITH KEY2 TO BE FOUND ON DISK
10969                                   ;X= 1  MUST BE ONE
10970                                   ;NOWORD= WITH NUMBER OF DATA WORDS TO BE WRITTEN
10971                                   ;THE COMMAND THEN IS JSR PC,COMHD
10972                                   ;
10973                                   ;
10974                                   ;
10975                                   ;
10976                                   ;
10977                                   ;
10978                                   ;IN A READ HEADER AND DATA COMMAND FILL THE FOLLOWING
10979                                   ;CYL= WITH CYLINDER TO BE FOUND ON DISK
10980                                   ;SECOTR= WITH SECTOR AND TRACK TO BE FOUND ON DISK
10981                                   ;KEY1= WITH KEY1 TO BE FOUND ON DISK
10982                                   ;KEY2=WITH KEY2 TO BE FOUND ON DISK
10983                                   ;DAWORD= WITH NUMBER OF WORDS TO BE FOUND ON DISK
10984                                   ;X=0 MUST BE ZERO
10985                                   ;THE COMMAND THEN IS JSR PC,COMHD
10986                                   ;
10987                                   ;
10988                                   ;
10989                                   ;
10990                                   ;
10991                                   ;
10992                                   ;
10993                                   ;
10994                                   ;
10995                                   ;
10996                                   ;IN A READ DATA COMMAND FILL THE FOLLOWING
10997                                   ;CYL= WITH CYLINDER TO BE FOUND ON DISK
10998                                   ;SECOTR= WITH SECTOR AND TRACK TO BE FOUND ON DISK
10999                                   ;KEY1= WITH KEY1 TO BE FOUND ON DISK
11000                                   ;KEY2=WITH KEY2 TO BE FOUND ON DISK
11001                                   ;DAWORD= WITH NUMBER OF WORDS TO BE FOUND ON DISK
11002                                   ;X=0 MUST BE ZERO
11003                                   ;THE COMMAND THEN IS JSR PC,COMHD
11004                                   ;
```

```
11005
11006
11007
11008
11009
11010
11011
11012                                   ;WRITE DATA COMMAND
11013                                   ;OR READ COMMAND I.E DATA ONLY OR HEADER AND DATA
11014  044302  011637  002000           COMHD:  MOV     (SP),@#PCJSR    ;SAVE PC OF JSR + 4
11015  044306  162737  000004  002000           SUB     #4,@#PCJSR              ;SAVE PC OF JSR
11016  044314  010046                           MOV     R0,-(SP)        ;;PUSH R0 ON STACK
11017  044316  010146                           MOV     R1,-(SP)        ;;PUSH R1 ON STACK
11018  044320  010246                           MOV     R2,-(SP)        ;;PUSH R2 ON STACK
11019  044322  010346                           MOV     R3,-(SP)        ;;PUSH R3 ON STACK
11020  044324  010446                           MOV     R4,-(SP)        ;;PUSH R4 ON STACK
11021  044326  010546                           MOV     R5,-(SP)        ;;PUSH R5 ON STACK
11022  044330  012777  000001  135312           MOV     #DMD,@RHMR      ;SET DIAGNOSTIS MODE
11023  044336  052777  000001  135264           BIS     #GO,@RHCS1      ;GO
11024  044344  016746  000044                   MOV     SECOTR, -(SP)   ;GET DESIRED SECTOR/TRACK
11025  044350  042716  177740                   BIC     #177740,        (SP)    ;MAKE ONLY SECTOR
11026  044354  012637  044364                   MOV     (SP)+,  @#TRK   ;SAVE SECTOR
11027  044360  004137  050426                   JSR     R1,     @#SEARCH        ;DO SEARCH SECTOR
11028  044364  000000                   TRK:    .WORD   0
11029  044366  012701  000240                   MOV     #+NOP,R1                ;GOING TO MOVE NOPS
11030  044372  010137  044424                   MOV     R1,@#SSYN       ;NOP INTO SSYN
11031  044376  010137  044426                   MOV     R1,@#HEDGAP     ;NOP INTO HEDGAP
11032  044402  010137  044430                   MOV     R1,@#HEDSYN     ;NOP INTO HEDSYN
11033  044406  004137  044534                   JSR     R1,     @#RDHEAD
11034  044412  000000                   CYL:    .WORD   0               ;CYLINDER ADDRESS
11035  044414  000000                   SECOTR: .WORD   0               ;SECTOR/TRACK ADDRESS
11036  044416  000000                   KEY1:   .WORD   0               ;KEY1 WORD
11037  044420  000000                   KEY2:   .WORD   0               ;KEY2 WORD
11038  044422  000000                   X:      .WORD   0               ;X=1 WRITE COMMAND
11039                                                                   ;X=0 READ COMMAND
11040  044424  000240                   SSYN:   NOP                     ;IF "ERROR 2" INSERTED BY RDHEAD
11041                                                                   ;SUBROUTINE THEN THE FIRST SYNC.
11042                                                                   ;IS NOT DETECTED.  NO BAD DATA
11043                                                                   ;IS GIVEN BECAUSE SYNC=144000
11044                                                                   ;CANNOT BE READ.  WORD NO
11045                                                                   ;IS "1" BECAUSE THIS IS THE FIRST
11046                                                                   ;WORD TESTED
11047
11048
11049
11050  044426  000240                   HEDGAP: NOP                     ;IF "ERROR 3" INSERTED BY
11051                                                                           ;RDHEAD SUBROUTINE THEN THE
11052                                                                   ;HEADER GAP 0'S WERE NOT
11053                                                                   ;WRITTEN RIGHT.
11054                                                                   ;IF "WORD NO" CONTAINS SAY
11055                                                                   ;3(8) THEN IT IS THE THIRD
11056                                                                   ;WORD OF A 5 WORD HEADER
11057                                                                   ;GAP THAT IS WRONG
11058                                                                   ;"BAD DATA" CONTAINS WHAT IS
11059                                                                   ;GOING ON THE DISK
11060
```

```
11061  044430  000240                  HEDSYN: NOP
11062                                                    ;IF "ERROR 3" INSERTED BY RDHEAD
11063                                                   ;SUBROUTINE THEN THE HEADER SYNC.
11064                                                   ;GENERATED BY DCL IS WRONG
11065                                                   ;OR THE LAST BYTE
11066                                                   ;OF THE HEADER GAP 0'S IS WRONG
11067                                                   ;IN EITHER CASE WORD NO=6
11068                                                   ;RIGHT BYTE IS HEADER 0
11069                                                   ;LEFT BYTE IS SYNC
11070                                                   ;"BAD DATA" HAS WHAT IS GOING
11071                                                   ;ON DISK
11072
11073  044432  005737  001772                  TST     @#ERFLG$                ;ARE ANY ERRORS DETECTED
11074  044436  001017                          BNE     OUT                     ;IF YES BRANCH
11075  044440  005737  044422                  TST     @#X
11076  044444  001410                          BEQ     DAREAD
11077  044446  005737  044526                  TST     @#NOSYNC                ;IS THIS FORCED HEADER ERROR COMMAND
11078                                                   ;IF YES  NOSYNC=-1 THEN WRITE OR READ
11079                                                   ;IS SHUT OFF SO BRANCH OUT
11080                                                   ;IF  NOSYNC=0 THEN CONTINUE
11081  044452  001011                          BNE     OUT                     ;BRANCH IF SET
11082  044454  004137  046000                  JSR     R1,     @#WRDATA        ;WRITE DATA
11083  044460  000000          NOWORD: .WORD   0                               ;NO OF WORDS TO BE WRITTEN
11084  044462  000000          Y:      .WORD   0                               ;
11085  044464  000404                  BR      OUT                             ;
11086  044466  004137  050702  DAREAD: JSR     R1,     @#REDATA        ;READ DATA
11087  044472  000000          DAWORD: .WORD   0                               ;NO OF WORDS TO BE READ
11088  044474  000000                  .WORD   0
11089  044476                  OUT:
11090  044476  012605                  MOV     (SP)+,R5                ;;POP STACK INTO R5
11091  044500  012604                  MOV     (SP)+,R4                ;;POP STACK INTO R4
11092  044502  012603                  MOV     (SP)+,R3                ;;POP STACK INTO R3
11093  044504  012602                  MOV     (SP)+,R2                ;;POP STACK INTO R2
11094  044506  012601                  MOV     (SP)+,R1                ;;POP STACK INTO R1
11095  044510  012600                  MOV     (SP)+,R0                ;;POP STACK INTO R0
11096  044512  000207                  RTS     PC
11097                                  ;;*****************************************************************
11098                                  ;THE DISK SECTOR IS DEVIDED AS FOLLOWS
11099                                  ;19 WORDS OF 0, ONE WORD 144000
11100                                  ;THESE MAKE 39 BYTES FOR SECTOR GAP AND ONE SYNC. BYTE
11101
11102
11103  044514  014400          RSYNC:  14400
11104  044516  000000          RCYL:   0
11105  044520  000000          RSETR:  0
11106  044522  000000          RKEY1:  0
11107  044524  000000          RKEY2:  0
11108
11109
11110                                  ;5 WORDS OF 0 ONE WORD 144000
11111                                  ;THESE MAKE 11 BYTES FOR HEADER GAP AND ONE SYNC. BYTE
11112                                  ;THESE ARE DCL GENERATED
11113
11114
11115
11116                                  ;THERE ARE 256 WORDS OF DATA
```

```
                                        ;THERE ARE 2 WORDS FOR ECC GENERATED BY DCL
                                        ;15 WORDS OF 0 FOR DATA GAP AND TOLERANCE GAP
                                        ;;*****************************************************************
                                        ;;*****************************************************************
                                        ;READ DISK HEADER
                                        ;;*****************************************************************
044526  000000                NOSYNC: 0                       ;FORCED HEADER ERROR = -1
                                                            ;NORMAL = 0
044530  000000                TY:     0                       ;ERROR TYPE NO.
044532  000000                ERWORD: 0                       ;ERROR WORD NO.
044534  012137  044516        RDHEAD: MOV     (R1)+,  @#RCYL  ;STORE CYLINDER ADDRESS
044540  012137  044520                MOV     (R1)+,  @#RSETR ;STORE SECTOR AND TRACK ADDRESS
044544  012137  044522                MOV     (R1)+,  @#RKEY1 ;STORE KEY1
044550  012137  044524                MOV     (R1)+,  @#RKEY2 ;STORE KEY2
044554  012137  045324                MOV     (R1)+,  @#COMPA ;STORE COMPARE OR NOT
044560  010146                        MOV     R1,-(SP)        ;PUSH R1 ON STACK
044562  013700  001650                MOV     @#RHMR, R0      ;R0 CONTAINS MAINTANENCE REG.
044566  012705  000002                MOV     #2,     R5      ;R5 IS A COUNTER FOR WORDS
044572  012710  000001                MOV     #DMD,   @R0     ;DIAG. MODE
044576  052710  000010                BIS     #MSTCK,@R0      ;SET SECTOR FOR FIRST WORD
044602  052710  000002                BIS     #MCLK,@R0       ;SET CLOCK FOR FIRST WORD
044606  042710  000012                BIC     #MSTCK!MCLK,@R0 ;RESET SECTOR AND CLOCK
044612  000404                        BR      2$              ;BRANCH OVER GIVING SECTOR FOR FIRST TIME
044614  012710  000013        1$:     MOV     #MSTCK!MCLK!DMD,@R0;SET SECTOR, CLOCK, DIAG. MODE, RESET INDEX
044620  042710  000012                BIC     #MSTCK!MCLK,@R0 ;RESET SECTOR, CLOCK
044624  012702  000007        2$:     MOV     #7,     R2      ;R2 IS A COUNTER FOR BYTES
044630  052710  000002        3$:     BIS     #MCLK,  @R0     ;SET CLOCK
044634  042710  000002                BIC     #MCLK,  @R0     ;RESET CLOCK
044640  005302                        DEC     R2              ;BYTE COUNTER
044642  001372                        BNE     3$              ;BRANCH IF BYTE NOT COMPLETE
044644  005305                        DEC     R5              ;WORD COUNTER
044646  001362                        BNE     1$              ;BRANCH IF WORD NOT COMPLETE
044650  012702  000022                MOV     #18.,   R2      ;NO OF WORDS OF ZEROS
044654  005037  045322        4$:     CLR     @#WORD          ;READ 0
044660  004737  045326                JSR     PC,     @#READ  ;GO TO READ
044664  005302                        DEC     R2              ;COUNT
044666  001372                        BNE     4$
044670  013737  044514  045322        MOV     @#RSYNC,@#WORD  ;SYNC. WORD
044676  004737  045326                JSR     PC,     @#READ
044702  032710  001000                BIT     #DTSY,  @R0     ;SYNC. BYTE DETECTED?
044706  001012                        BNE     5$              ;BRANCH IF SYNC DETECTED
```

```
11173  044710  012737  000001  044532          MOV     #1,     @#ERWORD        ;ERROR WORD NO
11174  044716  013737  044514  001124          MOV     @#RSYNC,@#$GDDAT        ;SYNC WORD
11175  044724  012737  104002  044424          MOV     #104002,@#SSYN  ;INSERT "ERROR 2" IN SSYN
11176  044732  000571                          BR      13$             ;BRANCH OUT
11177  044734  013737  044516  045322  5$:     MOV     @#RCYL, @#WORD  ;SETUP CYLINDER
11178  044742  004737  045326                  JSR     PC,     @#READ  ;READ
11179  044746  013737  044520  045322          MOV     @#RSETR,@#WORD  ;SETUP SECTOR/TRACK
11180  044754  004737  045326                  JSR     PC,@#READ       ;READ
11181  044760  013737  044522  045322          MOV     @#RKEY1,@#WORD  ;SETUP KEY1
11182  044766  004737  045326                  JSR     PC,@#READ       ;READ
11183  044772  013737  044524  045322          MOV     @#RKEY2,@#WORD  ;SETUP KEY2
11184  045000  004737  045326                  JSR     PC,@#READ       ;READ
11185  045004  013737  046312  045322          MOV     @#WCRC,@#WORD   ;SETUP CRC
11186  045012  004737  045326                  JSR     PC,@#READ       ;READ
11187  045016  005737  002012                  TST     @#TESDTE        ;IS THIS A DRIVE TIMING ERROR
11188  045022  001135                          BNE     13$             ;BRANCH OUT IF YES
11189  045024  005737  045324                  TST     @#COMPA         ;IS THIS A READ OR WRITE COMMAND
11190  045030  001472                          BEQ     11$
11191  045032  012705  046314                  MOV     #HEGAP, R5      ;POINTER FOR HEADER GAP
11192  045036  012702  000005                  MOV     #5,     R2      ;NO OF WORDS OF ZEROS
11193  045042  012737  000006  044532  6$:     MOV     #6,@#ERWORD     ;ERROR WORD NO SET
11194  045050  004737  045560                  JSR     PC,@#WRITE      ;FOR HEADER GAP
11195  045054  005737  045556                  TST     @#WWORD         ;TEST WRITTEN WORD
11196  045060  001413                          BEQ     7$              ;BRANCH IF GOOD THAT IS 0
11197  045062  160237  044532                  SUB     R2,@#ERWORD     ;WORD NO IN ERROR
11198  045066  005037  001124                  CLR     @#$GDDAT                ;GOOD WORD SHOULD BE 0
11199  045072  013767  045556  134026          MOV     @#WWORD,        $BDDAT  ;BAD DATA
11200  045100  012737  104003  044426          MOV     #104003,@#HEDGAP        ;"ERROR 2" GOES IN HEDGAP
11201  045106  000503                          BR      13$             ;BRANCH OUT
11202  045110  013725  045556          7$:     MOV     @#WWORD,(R5)+   ;SAVE HEADER GAP
11203  045114  005302                          DEC     R2
11204  045116  001351                          BNE     6$
11205  045120  004737  045560                  JSR     PC,     @#WRITE ;WRITE HEADER (DATA) GAP SYNC
11206  045124  023737  044514  045556          CMP     @#RSYNC,@#WWORD
11207  045132  001426                          BEQ     10$
11208  045134  005737  044526                  TST     @#NOSYNC                ;IS THIS FORCED HEADER ERROR COMMAND
11209                                                                          ;IF YES  NOSYNC=-1 THEN WRITE OR READ
11210                                                                          ;IS SHUT OFF SO BRANCH OUT
11211                                                                          ;IF NO NOSYNC=0 THEN CONTINUE
11212  045140  001406                          BEQ     14$                     ;BRANCH IF TRUE ERROR
11213  045142  005737  045556                  TST     @#WWORD                 ;
11214  045146  001420                          BEQ     10$                     ;BRANCH IF GOOD
11215  045150  005037  001124                  CLR     @#$GDDAT                ;IT SHOULD BE ZERO
11216  045154  000403                          BR      15$             ;BRANCH TO TYPE ERROR
11217  045156  013737  044514  001124  14$:     MOV     @#RSYNC,@#$GDDAT        ;GOOD DATA
11218  045164  013737  045556  001126  15$:     MOV     @#WWORD,@#$BDDAT        ;BAD DATA
11219  045172  012737  000006  044532          MOV     #6,     @#ERWORD
11220  045200  012737  104003  044430          MOV     #104003,@#HEDSYN
11221  045206  000443                          BR      13$             ;BRANCH OUT
11222  045210  013725  045556          10$:    MOV     @#WWORD,(R5)+   ;SAVE DATA SYNC.
11223  045214  000440                          BR      13$
11224                          ;READ COMMAND START FROM HERE
11225  045216  012702  000005          11$:    MOV     #5,     R2
11226  045222  005067  000074          12$:    CLR     WORD
11227  045226  004767  000074                  JSR     PC,     READ    ;READ HEADER GAP
11228  045232  005302                          DEC     R2              ;IS 5 HEADER GAP ZEROS COMPLETE
```

```
11229  045234  001372                          BNE     12$             ;IF NOT BRANCH
11230  045236  013737  044514  045322          MOV     @#RSYNC,@#WORD  ;SYNC WORD
11231  045244  004767  000056                  JSR     PC,     READ    ;READ HEADER (DATA) SYNC)
11232  045250  005737  044526                  TST     @#NOSYNC
11233  045254  001404                          BEQ     16$             ;IF NOT ERROR COMMAND BRANCH
11234  045256  032710  001000                  BIT     #DTSY,@RD       ;SYNC. DETECTED
11235  045262  001415                          BEQ     13$             ;IF ZERO BRANCH OUT
11236  045264  000403                          BR      17$             ;IF NOT ZERO BRANCH TO ERROR
11237  045266  032710  001000          16$:    BIT     #DTSY,  @RD     ;SYNC. DETECTED?
11238  045272  001011                          BNE     13$             ;BRANCH IF YES
11239  045274  012737  000006  044532  17$:    MOV     #6,@#ERWORD     ;ERROR WORD NO.
11240  045302  013737  044514  001124          MOV     @#RSYNC,@#$GDDAT;SYNC WORD
11241  045310  012737  104002  044430          MOV     #104002,@#HEDSYN
11242  045316                          13$:
11243  045316  012601                          MOV     (SP)+,R1        ;;POP STACK INTO R1
11244  045320  000201                          RTS     R1
11245
11246
11247
11248
11249
11250
11251
11252
11253                           ;;*****************************************************************
11254                           ;READ ONE WORD IN "WORD"
11255                           ;;*****************************************************************
11256
11257
11258
11259
11260  045322  000000          WORD:   0
11261  045324  000000          COMPA:  0
11262
11263
11264
11265
11266  045326                  READ:
11267  045326  010246                          MOV     R2,-(SP)        ;;PUSH R2 ON STACK
11268  045330  012705  000002                  MOV     #2,     R5      ;WORD COUNTER
11269  045334  012710  000001                  MOV     #DMD,   @RD     ;SET DIAG. MODE
11270  045340  006037  045322                  ROR     @#WORD          ;CHECKING IF THERE IS A ONE
11271  045344  103002                          BCC     1$              ;IF NO ONE BRANCH
11272  045346  052710  000020                  BIS     #MRD,   @RD     ;SET BIT 4 IF DATA HAS ONE
11273  045352  012702  000007          1$:     MOV     #7,     R2      ;BYTE COUNTER
11274  045356  052710  000012                  BIS     #MSTCK!MCLK,    @RD     ;SET CLOCK,DATA IF ANY, SECTOR
11275  045362  005737  042526                  TST     @#TSECCG        ;IS THIS BIT TO GENERATE AND TEST ECC
11276  045366  001411                          BEQ     6$              ;BRANCH IF NO
11277  045370  032710  000020                  BIT     #MRD,@RD        ;IS DATA BIT A ONE
11278  045374  001404                          BEQ     5$              ;BRANCH IF DATA BIT IS 0
11279  045376  012737  177777  042520          MOV     #-1,@#ECDATA    ;ECC DATA BIT IS A ONE
11280  045404  000402                          BR      6$              ;BRANCH
11281  045406  005037  042520          5$:     CLR     @#ECDATA        ;ECC DATA BIT IS A 0
11282  045412  012746  000001          6$:     MOV     #DMD,   -(SP)   ;KEEP ONLY DIAG. MODE
11283  045416  006037  045322                  ROR     @#WORD          ;CHECKING IF THERE IS A ONE
11284  045422  103002                          BCC     2$              ;IF NO ONE BRANCH
```

```
11285  045424  012716  000021          MOV     #MRD!DMD,       (SP)    ;KEEP DATA AND DIAG. MODE
11286  045430  012610                  2$:     MOV     (SP)+,  @R0     ;PUT IN DATA,RESET CLOCK, SECTOR
11287  045432  005737  042526                  TST     @#TSECCG        ;IS ECC TO BE GENERATED FOR THIS BIT
11288  045436  001404                          BEQ     3$              ;BRANCH IF NO
11289  045440  005237  042540                  INC     @#DATENV        ;NUMBER OF CLOCKS GIVEN FOR DATA ENVELOPE
11290  045444  004737  042556                  JSR     PC,@#ECTEST     ;GO TO GENERATE AND TEST ECC
11291  045450  052710  000002          3$:     BIS     #MCLK,  @R0     ;SET CLOCK
11292  045454  005737  042526                  TST     @#TSECCG        ;IS THIS BIT TO GENERATE ECC
11293  045460  001411                          BEQ     8$              ;BRANCH IF NO
11294  045462  032710  000020                  BIT     #MRD,@R0        ;IS DATA BIT A ONE
11295  045466  001404                          BEQ     7$              ;BRANCH IF DATA BIT IS = 0
11296  045470  012737  177777  042520          MOV     #-1,@#ECDATA    ;ECC DATA BIT IS A ONE
11297  045476  000402                          BR      8$              ;BRANCH
11298  045500  005037  042520          7$:     CLR     @#ECDATA        ;ECC DATA BIT IS = 0
11299  045504  012746  000001          8$:     MOV     #DMD,   -(SP)   ;KEEP DIAG. MODE
11300  045510  006037  045322                  ROR     @#WORD          ;CHECKING IF THERE IS A ONE
11301  045514  103002                          BCC     4$              ;BRANCH IF NO ONE
11302  045516  012716  000021                  MOV     #MRD!DMD,(SP)   ;KEEP DIAG. MODE AND DATA
11303  045522  012610                  4$:     MOV     (SP)+,  @R0     ;SET DATA, DIAG. MODE, CLEAR CLOCK
11304  045524  005737  042526                  TST     @#TSECCG        ;IS THIS BIT TO GENERATE ECC
'11305  045530  001404                          BEQ     9$              ;BRANCH IF NO
11306  045532  005237  042540                  INC     @#DATENV        ;NUMBER OF CLOCKS GIVEN FOR DATA ENVELOPE
11307  045536  004737  042556                  JSR     PC,@#ECTEST     ;GO TO GENERATE AND TEST ECC
11308  045542  005302                  9$:     DEC     R2              ;BYTE COUNTER
11309  045544  001341                          BNE     3$              ;BRANCH IF ONE BYTE NOT COMPLETE
11310  045546  005305                          DEC     R5              ;WORD COUNTER
11311  045550  001300                          BNE     1$              ;BRANCH IF ONE WORD NOT COMPLETE
11312  045552  012602                          MOV     (SP)+,R2        ;;POP STACK INTO R2
11313  045554  000207                          RTS     PC
11314
11315
11316
11317
11318
11319
11320                          ;;*****************************************************************
11321                          ;WRITE ONE WORD WHICH COMES BACK IN "WWORD"
11322                          ;;*****************************************************************
11323
11324
11325
11326
11327
11328
11329  045556  000000          WWORD:  0
11330
11331
11332
11333
11334  045560                  WRITE:
11335  045560  010046                  MOV     R0,-(SP)        ;;PUSH R0 ON STACK
11336  045562  010246                  MOV     R2,-(SP)        ;;PUSH R2 ON STACK
11337  045564  010346                  MOV     R3,-(SP)        ;;PUSH R3 ON STACK
11338  045566  010546                  MOV     R5,-(SP)        ;;PUSH R5 ON STACK
11339  045570  012705  000002          MOV     #2,     R5      ;WORD COUNTER
11340  045574  012710  000001          MOV     #1,     @R0     ;SET DIAG. MODE
```

```
11341  045600  012702  000007          1$:   MOV   #7,      R2          ;BYTE COUNTER
11342  045604  012710  000013                MOV   #MSTCK!MCLK!DMD,@R0;SET SECTOR AND CLOCK
11343  045610  032710  000040                BIT   #MWR,     @R0         ;CHECK WRITEBIT IN MAINT. REG.
11344  045614  001406                        BEQ   2$                  ;BRANCH IF ZERO
11345  045616  012737  177777  042520        MOV   #-1,@#ECDATA        ;ECC DATA BIT IS A ONE
11346  045624  000261                        SEC                       ;SET CARRY
11347  045626  006003                        ROR   R3                  ;MOVE 1 FORWARD
11348  045630  000404                        BR    3$
11349  045632  005037  042520          2$:   CLR   @#ECDATA            ;ECC DATA BIT IS = 0
11350  045636  000241                        CLC                       ;CLEAR CARRY
11351  045640  006003                        ROR   R3                  ;MOVE 0 FOR WWORD
11352  045642  012710  000001          3$:   MOV   #DMD,     @R0         ;CLEAR SECTOR AND CLOCK
11353  045646  005737  042526                TST   @#TSECCG            ;IS THIS BIT TO GENERATE ECC
11354  045652  001404                        BEQ   4$                  ;BRANCH IF NO
11355  045654  005237  042540                INC   @#DATENV            ;NUMBER OF CLOCKS GIVEN FOR DATA ENVELOPE
11356  045660  004737  042556                JSR   PC,@#ECTEST         ;GO TO GENERATE AND TEST ECC
11357  045664  052710  000002          4$:   BIS   #MCLK,    @R0         ;SET CLOCK
11358  045670  032710  000040                BIT   #MWR,     @R0         ;CHECK WRITE BIT IN MAINT. REG.
11359  045674  001406                        BEQ   5$                  ;BRANCH IF ZERO
11360  045676  012737  177777  042520        MOV   #-1,@#ECDATA        ;ECC DATA BIT IS A ONE
11361  045704  000261                        SEC                       ;SET CARRY
11362  045706  006003                        ROR   R3                  ;MOVE 1 FOR WWORD
11363  045710  000404                        BR    6$
11364  045712  005037  042520          5$:   CLR   @#ECDATA            ;ECC DATA BIT IS ZERO
11365  045716  000241                        CLC                       ;CLEAR CARRY
11366  045720  006003                        ROR   R3                  ;MOVE 0 FOR WWORD
11367  045722  012710  000001          6$:   MOV   #DMD,     @R0         ;CLEAR CLOCK
11368  045726  005737  042526                TST   @#TSECCG            ;IS THIS BIT TO GENERATE ECC
11369  045732  001404                        BEQ   7$                  ;BRANCH IF NO
11370  045734  005237  042540                INC   @#DATENV            ;NUMBER OF CLOCKS GIVEN FOR DATA ENVELOPE
11371  045740  004737  042556                JSR   PC,@#ECTEST         ;GO TO GENERATE AND TEST ECC
11372  045744  005302                  7$:   DEC   R2                  ;COUNT FOR BYTE END
11373  045746  001346                        BNE   4$                  ;IF NOT BYTE END BRANCH
11374  045750  005305                        DEC   R5                  ;COUNT FOR WORD END
11375  045752  001312                        BNE   1$                  ;IF NOT WORD END BRANCH
11376  045754  010337  045556                MOV   R3,      @#WWORD    ;STORE WORD
11377  045760  012605                        MOV   (SP)+,R5            ;;POP STACK INTO R5
11378  045762  012603                        MOV   (SP)+,R3            ;;POP STACK INTO R3
11379  045764  012602                        MOV   (SP)+,R2            ;;POP STACK INTO R2
11380  045766  012600                        MOV   (SP)+,R0            ;;POP STACK INTO R0
11381  045770  000207                        RTS   PC
11382
11383
11384
11385
11386
11387
11388                          ;;*****************************************************************
11389                          ;WRITE DATA
11390                          ;;*****************************************************************
11391
11392
11393
11394
11395
11396
```

```
11397
11398  045772  000000                  COUNTD: 0
11399  045774  000400                  FORMAT: 256.
11400  045776  000000                  ZWORDS: 0
11401  046000                          WRDATA:
11402  046000  011137  045772                  MOV     (R1),   @#COUNTD        ;STORE NO. OF WORDS TO BE WRITTEN
11403  046004  012102                          MOV     (R1)+,  R2      ;SAME IN R2
11404  046006  012137  045324                  MOV     (R1)+,  @#COMPA ;COMPARE OR NOT
11405  046012  010046                          MOV     R0,-(SP)        ;;PUSH R0 ON STACK
11406  046014  010146                          MOV     R1,-(SP)        ;;PUSH R1 ON STACK
11407  046016  010246                          MOV     R2,-(SP)        ;;PUSH R2 ON STACK
11408  046020  010346                          MOV     R3,-(SP)        ;;PUSH R3 ON STACK
11409  046022  010446                          MOV     R4,-(SP)        ;;PUSH R4 ON STACK
11410  046024  012701  000016                  MOV     #14.,   R1      ;NO. OF TOLERANCE GAP WORDS
11411  046030  012703  047336                  MOV     #TOLGAP,R3      ;START OF TOLERANCE GAP TABLE
11412  046034  012723  177777          1$:     MOV     #-1,    (R3)+   ;MAKE IT 177777
11413  046040  005301                          DEC     R1              ;IS 14 COMPLETED
11414  046042  001374                          BNE     1$              ;IF NO BRANCH
11415  046044  013700  001650                  MOV     @#RHMR, R0      ;R0 CONTAINS MAINTANENCE REG.
11416  046050  013746  045774                  MOV     @#FORMAT,       -(SP)
11417  046054  163716  045772                  SUB     @#COUNTD,       (SP)
11418  046060  011637  045776                  MOV     (SP),@#ZWORDS   ;NO. OF ZERO WORDS TO BE WRITTEN
11419  046064  012604                          MOV     (SP)+,  R4
11420  046066  005737  002010                  TST     @#TSECC         ;IS THIS AN ECC TEST
11421  046072  001403                          BEQ     7$              ;BRANCH IF NO
11422  046074  012737  177777  042526          MOV     #-1,@#TSECCG    ;THESE BITS ARE TO GENERATE ECC
11423  046102  012703  046330          7$:     MOV     #DISK,  R3      ;SIMULATED DISK AREA
11424  046106  004737  045560          2$:     JSR     PC,     @#WRITE ;WRITE ON SIMULATED DISK
11425  046112  013723  045556                  MOV     @#WWORD,(R3)+   ;STORE ON SIMULATED DISK
11426  046116  005302                          DEC     R2
11427  046120  001372                          BNE     2$
11428  046122  005704                          TST     R4      ;ANY ZEROS TO BE WRITTEN
11429  046124  001406                          BEQ     4$      ;BRANCH IF NONE TO BE WRITTEN
11430  046126  004737  045560          3$:     JSR     PC,     @#WRITE ;WRITE ZEROS ON SIMULATED DISK
11431  046132  013723  045556                  MOV     @#WWORD,        (R3)+   ;STORE
11432  046136  005304                          DEC     R4
11433  046140  001372                          BNE     3$
11434  046142  005037  042526          4$:     CLR     @#TSECCG        ;NO MORE ECC TO BE GENERATED
11435  046146  012701  000002                  MOV     #2,     R1
11436  046152  004767  177402          5$:     JSR     PC,     WRITE   ;WRITE ECC1 AND ECC2 ON SIMULATED DISK
11437  046156  013723  045556                  MOV     @#WWORD,(R3)+   ;STORE ON WEEC1 AND WEEC2
11438  046162  005301                          DEC     R1
11439  046164  001372                          BNE     5$
11440  046166  004767  177366                  JSR     PC,     WRITE   ;WRITE DATA GAP
11441  046172  013723  045556                  MOV     @#WWORD,(R3)+   ;STORE
11442  046176  012701  000016                  MOV     #14.,   R1
11443  046202  004737  045560          6$:     JSR     PC,     @#WRITE ;WRITE TOLERANCE GAP ZEROS
11444  046206  013723  045556                  MOV     @#WWORD,(R3)+   ;STORE
11445  046212  005301                          DEC     R1
11446  046214  001372                          BNE     6$
11447  046216  012604                          MOV     (SP)+,R4        ;;POP STACK INTO R4
11448  046220  012603                          MOV     (SP)+,R3        ;;POP STACK INTO R3
11449  046222  012602                          MOV     (SP)+,R2        ;;POP STACK INTO R2
11450  046224  012601                          MOV     (SP)+,R1        ;;POP STACK INTO R1
11451  046226  012600                          MOV     (SP)+,R0        ;;POP STACK INTO R0
11452  046230  000201                          RTS     R1
```

```
11453
11454
11455
11456
11457
11458
11459
11460
11461                                   ;;*****************************************************************
11462                                   ;WRITE HEADER AND DATA
11463                                   ;
11464                                   ;
11465                                   ;THIS IS THE SIMULATED DISK
11466                                   ;ONLY ONE SECTOR OF SPACE IS ALLOWED
11467                                   ;;*****************************************************************
11468
11469
11470
11471
11472
11473
11474
11475  046232  000023                   SECGAP: .BLKW   19.             ;SECTOR GAP 38 BYTES OF 0
11476  046300  000001                   WSSYNC: .BLKW   1               ;SECTOR GAP 1 BYTE OF 0 ONE SYNC BYTE
11477  046302  000004                   HEADER: .BLKW   4               ;HEADER = CYL, SECTOR/TRACK, KEY1, KEY2
11478  046312  000001                   WCRC:   .BLKW   1               ;CRC
11479  046314  000005                   HEGAP:  .BLKW   5               ;HEADER GAP 10 BYTES OF 0
11480  046326  000001                   HDWSYN: .BLKW   1               ;HEADER GAP 1 BYTE OF 0 ONE SYNC. BYTE
11481  046330                           SILOTB:                 ;USED IN SILO TEST AS SILO TABLE
11482  046330  000400                   DISK:   .BLKW   256.            ;DATA SPACE
11483  047330  000001                   WECC1:  .BLKW   1               ;ECC1
11484  047332  000001                   WECC2:  .BLKW   1               ;ECC2
11485  047334  000001                   DTAGAP: .BLKW   1               ;DATA GAP 2 BYTES OF 0
11486  047336  000016                   TOLGAP: .BLKW   14.             ;TOLERANCE GAP 28 BYTES OF 0
11487
11488
11489
11490
11491
11492
11493                                   ;;*****************************************************************
11494                                   ;WRITE HEADER AND DATA
11495                                   ;;*****************************************************************
11496
11497
11498
11499
11500  047372  011637  002000           COMWHD: MOV     (SP),@#PCJSR    ;SAVE PC OF JSR + 4
11501  047376  162737  000004  002000           SUB     #4,@#PCJSR              ;SAVE PC OF JSR
11502  047404  010046                           MOV     R0,-(SP)        ;;PUSH R0 ON STACK
11503  047406  010146                           MOV     R1,-(SP)        ;;PUSH R1 ON STACK
11504  047410  010246                           MOV     R2,-(SP)        ;;PUSH R2 ON STACK
11505  047412  010346                           MOV     R3,-(SP)        ;;PUSH R3 ON STACK
11506  047414  010446                           MOV     R4,-(SP)        ;;PUSH R4 ON STACK
11507  047416  010546                           MOV     R5,-(SP)        ;;PUSH R5 ON STACK
11508  047420  012777  000001  132222           MOV     #DMD,@RHMR              ;SET DIAGNOSTIC MODE
```

```
11509  047426  052777  000001  132174          BIS     #GO,@RHCS1          ;GO
11510  047434  013746  047520                  MOV     @#WSECTR,   -(SP)   ;GET DESIRED SECTOR/TRACK
11511  047440  042716  177740                  BIC     #177740,    (SP)    ;MAKE ONLY SECTOR
11512  047444  012637  047454                  MOV     (SP)+,  @#WTRK  ;SAVE SECTOR
11513  047450  004137  050426                  JSR     R1,@#SEARCH     ;DO SEARCH SECTOR
11514  047454  000000                  WTRK:   .WORD   0               ;SECTOR NO.
11515  047456  012701  000240                  MOV     #+NOP,R1        ;GOING TO MOVE NOPS
11516  047462  010137  047530                  MOV     R1,@#SEGPER     ;NOP INTO SEGAP
11517  047466  010137  047532                  MOV     R1,@#FSYNER     ;NOP INTP FSYNER
11518  047472  010137  047534                  MOV     R1,@#ERHEAD     ;NOP INTO ERHEAD
11519  047476  010137  047536                  MOV     R1,@#ERCRC      ;NOPINTO ERCRC
11520  047502  010137  047540                  MOV     R1,@#ERHDGP     ;NOP INTO ERHDGAP
11521  047506  010137  047542                  MOV     R1,@#HDESYN     ;NOP INTO HDESYN
11522  047512  004137  047612                  JSR     R1,@#WRHEAD
11523  047516  000000                  WCYL:   0               ;CYLINDER
11524  047520  000000                  WSECTR: 0               ;SECTOR AND TRACK
11525  047522  000000                  WKEY1:  0               ;KEY1
11526  047524  000000                  WKEY2:  0               ;KEY2
11527  047526  000000                  GCRC:   0               ;GOOD CRC
11528
11529  047530  000240                  SEGPER: NOP             ;IF "ERROR 6" INSERTED BY
11530                                                          ;WRHEAD SUBROUTINE THEN
11531                                                          ;SECTOR GAP GOING ON DISK
11532                                                          ;IS NOT RIGHT
11533                                                          ;WORD NO. CONTAINS WHICH
11534                                                          ;WORD IS WRONG THAT IS
11535                                                          ;FIRST OF TENTH OR WHAT EVER NO.
11536                                                          ;BAD WORD IS WHAT IS GOING ON DISK
11537  047532  000240                  FSYNER: NOP             ;IF "ERROR 6" INSERTED BY
11538                                                          ;WRHEAD SUBROUTINE THEN
11539                                                          ;THE LAST 0 BYTE OF SECTOR
11540                                                          ;GAP OF FIRST SYNC. BYTE
11541                                                          ;AFTER SECTOR GAP IS IN
11542                                                          ;ERROR
11543                                                          ;WORD NO. CONTAINS 20
11544                                                          ;RIGHT BYTE IS SECTOR GAP
11545                                                          ;LEFT BYTE IS SYNC. BYTE
11546                                                          ;BAD WORD IS WHAT IS GOING ON
11547                                                          ;DISK
11548
11549  047534  000240                  ERHEAD: NOP
11550                                                          ;IF "ERROR 6" INSERTED BY
11551                                                          ;WRHEAD SUBROUTINE THEN
11552                                                          ;HEADER GOING ON DISK
11553                                                          ;IS WRONG
11554                                                          ;WORD NO 1 = CYLINDER NO
11555                                                          ;WORD NO 2 = SECTOR/TRACK
11556                                                          ;WORD NO 3 = KEY1
11557                                                          ;WORD NO 4 = KEY2
11558                                                          ;BAD WORD IS WHAT IS GOING ON
11559                                                          ;DISK
11560
11561  047536  000240                  ERCRC:  NOP
11562                                                          ;IF "ERROR 6" INSERTED BY
11563                                                          ;WRHEAD SUBROUTINE THEN CRC WRITTEN
11564                                                          ;ON DISK IS IN ERROR
```

```
11565
11566
11567
11568  047540  000240                    ERHDGP: NOP                     ;GOOD DATA IS WHAT SHOULD BE ON DISK
11569                                                                     ;BAD DATA IS WHAT IS GOING ON DISK
11570                                                                     ;WORD NO IS 5
11571                                                                     ;IF "ERROR 6" INSERTED BY
11572                                                                     ;WRHEAD SUBROUTINE THEN HEADER
11573                                                                     ;GAP GOING ON DISK IS WRONG
11574                                                                     ;WORD NO GIVES WHICH OF
11575                                                                     ;THE HEADER GAP WORDS
11576                                                                     ;ARE WRONG. FOR EXAMPLE
11577                                                                     ;WORD NO 1 = FIRST HEADER
11578                                                                     ;           GAP WORD
11579  047542  000240                    HDESYN: NOP                     ;BAD WORD IS WHAT OS GOING ON DISK
11580
11581                                                                     ;IF "ERROR 6" INSERTED BY
11582                                                                     ;WRHEAD SUBROUTINE THEN LAST
11583                                                                     ;HEADER GAP BYTF OR HEADER
11584                                                                     ;SYNC BYTE GOING ON DISK IS WRONG
11585                                                                     ;WORD NO = 5
11586                                                                     ;BAD DATA IS WHAT IS GOING
11587                                                                     ;ON DISK RIGHT BYTE IS HEADER
11588                                                                     ;GAP 0 BYTE, LEFT BYTE IS HEADER
11589                                                                     ;GAP SYNC.
11590
11591  047544  005737  001772                    TST     @#ERFLG$                ;ARE ANY ERRORS DETECTED
11592  047550  001004                            BNE     FOUT            ;IF YES BRANCH
11593  047552  004137  046000                    JSR     R1,@#WRDATA
11594  047556  000000                    FNWORD: .WORD   0               ;FORMAT COMMAND NO. OF DATA
11595  047560  000000                            .WORD   0
11596  047562                            FOUT:
11597  047562  012605                            MOV     (SP)+,R5        ;;POP STACK INTO R5
11598  047564  012604                            MOV     (SP)+,R4        ;;POP STACK INTO R4
11599  047566  012603                            MOV     (SP)+,R3        ;;POP STACK INTO R3
11600  047570  012602                            MOV     (SP)+,R2        ;;POP STACK INTO R2
11601  047572  012601                            MOV     (SP)+,R1        ;;POP STACK INTO R1
11602  047574  012600                            MOV     (SP)+,R0        ;;POP STACK INTO R0
11603  047576  000207                            RTS     PC
11604
11605
11606
11607
11608
11609
11610                                   ;;*****************************************************************
11611                                   ;WRITE HEADER
11612                                   ;;*****************************************************************
11613
11614
11615
11616
11617
11618                                   ;R0 = MAINT.REG.; R1 = SIMULATED DISK; R2 = BYTE COUNT; R3 = WRITE WORD; R5 = WORD COUNT
11619
11620
```

```
11621
11622  047600  000000                  SCYL:   0                       ;
11623  047602  000000                  SSECTR: 0                       ;
11624  047604  000000                  SKEY1:  0                       ;
11625  047606  000000                  SKEY2:  0                       ;
11626  047610  000000                  SCRC:   0                       ;
11627
11628
11629  047612  012137  047600          WRHEAD: MOV     (R1)+,  @#SCYL
11630  047616  012137  047602                  MOV     (R1)+,  @#SSECTR
11631  047622  012137  047604                  MOV     (R1)+,  @#SKEY1
11632  047626  012137  047606                  MOV     (R1)+,  @#SKEY2
11633  047632  012137  047610                  MOV     (R1)+,  @#SCRC
11634  047636  010146                          MOV     R1,-(SP)        ;;PUSH R1 ON STACK
11635  047640  012701  046232                  MOV     #SECGAP,R1      ;SIMULATED DISK INDICATOR
11636  047644  013700  001650                  MOV     @#RHMR, R0      ;R0 NOW HAS MAINT. REG. ADDR.
11637  047650  012710  000001                  MOV     #DMD,   @R0     ;SET DIAG. MODE
11638  047654  012705  000002                  MOV     #2,     R5      ;WORD COUNTER
11639  047660  052710  000010                  BIS     #MSTCK,@R0      ;SET SECTOR FOR FIRST BYTE
11640  047664  012710  000013  1$:             MOV     #MSTCK!MCLK!DMD,@R0;SET SECTOR, CLOCK, DIAG. MODE, RESET INDEX
11641  047670  032710  000040                  BIT     #MWR,   @R0     ;CHECK WRITE BIT IN MAINT. REG.
11642  047674  001403                          BEQ     2$
11643  047676  000261                          SEC                     ;SET CARRY
11644  047700  006003                          ROR     R3              ;MOVE ONE FORWARD
11645  047702  000402                          BR      3$              ;
11646  047704  000241  2$:             CLC                             ;CLEAR CARRY
11647  047706  006003                          ROR     R3              ;MOVE ZERO FORWARD
11648  047710  012710  000001  3$:             MOV     #DMD,   @R0     ;CLEAR CLOCK, SECTOR
11649  047714  012702  000007                  MOV     #7,     R2      ;BYTE COUNTER
11650  047720  052710  000002  4$:             BIS     #MCLK,  @R0     ;SET CLOCK
11651  047724  032710  000040                  BIT     #MWR,   @R0     ;CHECK WRITE BIT IN MAINT.REG.
11652  047730  001403                          BEQ     5$              ;BRANCH IF ZERO
11653  047732  000261                          SEC                     ;SET CARRY
11654  047734  006003                          ROR     R3              ;MOVE ONE FORWARD
11655  047736  000402                          BR      6$
11656  047740  000241  5$:             CLC
11657  047742  006003                          ROR     R3
11658  047744  012710  000001  6$:             MOV     #DMD,   @R0
11659  047750  005302                          DEC     R2
11660  047752  001362                          BNE     4$
11661  047754  005305                          DEC     R5
11662  047756  001342                          BNE     1$
11663  047760  010321                          MOV     R3,     (R1)+
11664  047762  005703                          TST     R3
11665  047764  001414                          BEQ     7$
11666  047766  012737  000001  044532          MOV     #1,@#ERWORD
11667  047774  005037  001124                  CLR     @#$GDDAT
11668  050000  010337  001126                  MOV     R3,     @#$BDDAT
11669  050004  012737  104006  047530          MOV     #104006,@#SEGPER
11670  050012  000137  050420                  JMP     @#17$           ;BRANCH OUT
11671  050016  012702  000022  7$:             MOV     #18.,   R2      ;COUNT NO. OF SECTOR GAP
11672  050022  012737  000024  044532  10$:    MOV     #20.,@#ERWORD   ;COUNT TO GIVE ERROR WORD
11673  050030  004737  045560                  JSR     PC,     @#WRITE ;WRITE SECTOR GAP
11674  050034  013721  045556                  MOV     @#WWORD,(R1)+   ;STORE SECTOR GAP WORD
11675  050040  001413                          BEQ     11$
11676  050042  160237  044532                  SUB     R2,@#ERWORD     ;IF NOT GET ERROR WORD NO.
```

```
11677  050046  005037  001124                      CLR     @#$GDDAT                ;GOOD WORD
11678  050052  013737  045556  001126              MOV     @#WWORD,@#$BDDAT        ;BAD WORD
11679  050060  012737  104006  047530              MOV     #104006,@#SEGPER        ;STORE "ERROR 6" IN SEGPER
11680  050066  000554                              BR      17$             ;BRANCH OUT
11681  050070  005302                      11$:    DEC     R2              ;HAVE 18 WORDS OF ZEROS BEEN WRITTEN
11682  050072  001353                              BNE     10$             ;IF NOT BRANCH
11683                                               ;AT THIS POINT THE SECTOR FOUND FLOP SHOULD
11684                                               ;BE HIGH.   SO THAT THE HEADER SYNC BYTE CAN BE GIVEN
11685                                               ;HOWEVER IN THE DRIVE TIMING ERROR TEST THE REST OF THE ROUTINE
11686                                               ;IS ABORTED
11687  050074  005737  002012                      TST     @#TESDTE        ;IS THIS A DRIVE TIMING ERROR
11688  050100  001147                              BNE     17$             ;BRANCH OUT IF YES
11689  050102  004737  045560                      JSR     PC,     @#WRITE ;WRITE ONE SECTOR GAP 0 BYTE
11690                                                                       ;AND ONE SYNC. BYTE = 230
11691  050106  013711  045556                      MOV     @#WWORD,(R1)    ;SAVE 0 BYTE AND SYNC BYTE
11692  050112  023721  044514                      CMP     @#RSYNC,(R1)+   ;IF SYNC. BYTE RIGHT
11693  050116  001414                              BEQ     12$             ;IF YES BRANCH
11694  050120  012737  000024  044532              MOV     #20.,@#ERWORD   ;IF NOT GET READY FOR ERROR
11695  050126  013737  044514  001124              MOV     @#RSYNC,@#$GDDAT        ;GOOD WORD
11696  050134  014137  001126                      MOV     -(R1),@#$BDDAT  ;BAD WORD
11697  050140  012737  104006  047532              MOV     #104006,@#FSYNER        ;INSERT "ERROR 6" IN FSYNER
11698  050146  000524                              BR      17$             ;BRANCH OUT
11699  050150  012702  000004              12$:    MOV     #4,     R2      ;FOUR HEADER WORDS
11700  050154  012703  047600                      MOV     #SCYL,  R3      ;POINTER FOR HEADER TABLE
11701  050160  012737  000005  044532      13$:    MOV     #5,@#ERWORD     ;ERROR WORD NO SET
11702  050166  004737  045560                      JSR     PC,     @#WRITE ;WRITE 4 HEADER WORDS
11703  050172  013711  045556                      MOV     @#WWORD,(R1)    ;STORE WRITTEN WORD
11704  050176  022321                              CMP     (R3)+,  (R1)+   ;IS IT RIGHT?
11705  050200  001412                              BEQ     14$             ;IF GOOD BRANCH
11706                                                                       ;IF NOT GET READY FOR PRINT
11707  050202  160237  044532                      SUB     R2,@#ERWORD     ;WORD NO
11708  050206  014337  001124                      MOV     -(R3),@#$GDDAT  ;GOOD DATA
11709  050212  014137  001126                      MOV     -(R1),@#$BDDAT  ;BAD DATA
11710  050216  012737  104006  047534              MOV     #104006,@#ERHEAD        ;INSERT "ERROR 6"
11711  050224  000475                              BR      17$             ;BRANCH OUT
11712  050226  005302                      14$:    DEC     R2              ;ARE 4 HEADER WORDS DONE?
11713  050230  001353                              BNE     13$             ;IF NOT BRANCH
11714  050232  004737  045560                      JSR     PC,     @#WRITE ;WRITE CRC
11715  050236  013711  045556                      MOV     @#WWORD,(R1)    ;STORE CRC
11716  050242  022137  047526                      CMP     (R1)+,@#GCRC    ;COMPARE GOOD CRC
11717  050246  001414                              BEQ     20$             ;BRANCH IF GOOD
11718  050250  014137  001126                      MOV     -(R1),@#$BDDATA ;BAD CRC WRITTEN
11719  050254  013737  047526  001124              MOV     @#GCRC,@#$GDDAT ;GOOD CRC
11720  050262  012737  000005  044532              MOV     #5,@#ERWORD     ;ERROR WORD NO
11721  050270  012737  104006  047536              MOV     #104006,@#ERCRC ;INSERT ERROR 6
11722  050276  000450                              BR      17$
11723  050300  012702  000005              20$:    MOV     #5,     R2      ;NO OF HEADER GAP
11724  050304  012737  000006  044532      15$:    MOV     #6,@#ERWORD     ;ERROR WORD NO SET
11725  050312  004737  045560                      JSR     PC,@#WRITE      ;WRITE HEADER GAP
11726  050316  013721  045556                      MOV     @#WWORD,(R1)+   ;STORE
11727  050322  001412                              BEQ     16$             ;IF GOOD BRANCH
11728  050324  160237  044532                      SUB     R2,@#ERWORD     ;ERROR WORD NO
11729  050330  005037  001124                      CLR     @#$GDDAT                ;GOOD DATA
11730  050334  014137  001126                      MOV     -(R1),@#$BDDAT  ;BAD DATA
11731  050340  012737  104006  047540              MOV     #104006,@#ERHDGP        ;STORE "ERROR 6"
11732  050346  000424                              BR      17$             ;BRANCH OUT
```

```
11733  050350  005302                  16$:    DEC     R2              ;ARE 5 HEADER GAP ZEROS  DONE
11734  050352  001354                          BNE     15$             ;IF NOT BRANCH
11735  050354  004737  045560                  JSR     PC,@#WRITE
11736  050360  013711  045556                  MOV     @#WWORD,(R1)
11737  050364  023721  044514                  CMP     @#RSYNC,(R1)+
11738  050370  001413                          BEQ     17$
11739  050372  012737  000005  044532          MOV     #5,@#ERWORD
11740  050400  014137  001126                  MOV     -(R1),  @#$BDDAT
11741  050404  013737  044514  001124          MOV     @#RSYNC,        @#$GDDAT
11742  050412  012737  104006  047542          MOV     #104006,@#HDESYN
11743  050420                          17$:
11744  050420  012601                          MOV     (SP)+,R1        ;;POP STACK INTO R1
11745  050422  000201                          RTS     R1
11746
11747
11748
11749
11750
11751
11752
11753
11754                                   ;;*****************************************************************
11755                                   ;SEARCH SECTOR
11756                                   ;;*****************************************************************
11757
11758
11759
11760
11761
11762
11763                                   ;       R0=RHMR ADDRESS
11764                                   ;       R1=PASSED ARGUMENT (SECTOR SEARCHED FOR)
11765                                   ;       R2=CLOCK COUNT (PER BYTE)
11766                                   ;       R3=SECTOR COUNTER FROM R1
11767                                   ;       R5=BYTES PER WORD COUNT
11768                                   ;BEBORE INDEX IS GIVEN TWO SECTOR CLOCKS ARE GIVEN TO RESET
11769                                   ;SECTOR PULSE IN CASE IT IS SET
11770                                   ;AT BEGINNING OF EACH SECTOR ONE SECTOR CLOCK HAS TO RISE
11771                                   ;BEFORE CLOCK THEN EVERY EIGHT CLOCKS ONE SECTOR CLOCK IS
11772                                   ;IDENTICAL WITH CLOCK
11773                                   ;NUMBERING THE SECTOR CLOCKS AS FOLLOWS
11774                                   ;THE SECTOR CLOCK UNDER INDEX - 0
11775                                   ;THE NEXT - 1
11776                                   ;THE NEXT  - 2
11777                                   ;ETC.
11778                                   ;THEN THE LAST SECTOR CLOCK IN ONE SECTOR HAS NUMBER - 608
11779                                   ;THE NEXT SECTOR THEN HAS 608 SECTOR CLOCKS
11780                                   ;THE NEXT SECTOR THEN HAS ANOTHER 608 SECTOR CLOCKS
11781                                   ;AND SO ON
11782
11783
11784
11785  050424  000000                  SECTR:  0                       ;SECTOR SEARCHED FOR
11786
11787  050426  012137  050424          SEARCH: MOV     (R1)+,  @#SECTR ;SAVE SECTOR SEARCHED FOR
11788  050432  010046                          MOV     R0,-(SP)        ;;PUSH R0 ON STACK
```

```
11789  050434  010146                        MOV     R1,-(SP)          ;;PUSH R1 ON STACK
11790  050436  010246                        MOV     R2,-(SP)          ;;PUSH R2 ON STACK
11791  050440  010346                        MOV     R3,-(SP)          ;;PUSH R3 ON STACK
11792  050442  010446                        MOV     R4,-(SP)          ;;PUSH R4 ON STACK
11793  050444  010546                        MOV     R5,-(SP)          ;;PUSH R5 ON STACK
11794  050446  013700  001650                MOV     @#RHMR, R0        ;NOW R0 HAS MAINTENANCE REG. ADR.
11795  050452  013703  050424                MOV     @#SECTR,R3        ;SECTOR COUNTER
11796  050456  012710  000001                MOV     #DMD,   @R0       ;SET DIAGNOSTIC MODE
11797  050462  052710  000010                BIS     #MSTCK, @R0       ;SET SECTOR CLOCK
11798  050466  042710  000010                BIC     #MSTCK, @R0       ;CLEAR SECTOR CLOCK
11799  050472  052710  000010                BIS     #MSTCK, @R0       ;SET SECTOR CLOCK
11800  050476  042710  000010                BIC     #MSTCK, @R0       ;CLEAR SECTOR CLOCK
11801                                                                    ;THE ABOVE TWO SECTOR CLOCKS ARE GIVEN FOR
11802                                                                    ;RESETING SECTOR PULSE
11803                                                                    ;IN CASE IT STARTS SET
11804  050502  052710  000014                BIS     #MINX!MSTCK,@R0   ;SET INDEX AND SECTOR CLOCK
11805  050506  012710  000001                MOV     #DMD,   @R0       ;RESET INDEX AND SECTOR CLOCK
11806  050512  005703                        TST     R3                ;IF SECTOR REQUIRED JUMP OUT
11807  050514  001461                        BEQ     7$                ;BRANCH OF SECTOR ZERO REQUIRED
11808                                        ;NOW THE 304 WORDS WILL START
11809
11810
11811
11812                          ;FOR FIRST BYTE SECTOR CLOCK WILL GO HIGH THEN CLOCK WILL GO HIGH
11813                          ;BOTH WILL COME DOWN TOGETHER THEN SEVEN CLOCKS WILL BE GIVEN
11814                          ;;FOR SECOND BYTE AND ALL OTHER BYTES TILL NEXT SECTOR SECTOR CLOCK
11815                          ;WILL BE IDENTICAL WITH ONE CLOCK
11816
11817
11818                                        ;ONE WORD ONLY
11819
11820  050516  012702  000010          1$:   MOV     #8.,    R2        ;BYTE COUNTER
11821  050522  012705  000002                MOV     #2,     R5        ;BYTES PER WORD
11822  050526  052710  000010                BIS     #MSTCK,@R0        ;SET SECTOR CLOCK
11823  050532  052710  000002                BIS     #MCLK,@R0         ;SET CLOCK
11824  050536  000402                        BR      3$                ;BRANCH TO CLEAR SECTOR AND CLOCK
11825  050540  052710  000012          2$:   BIS     #MSTCK!MCLK,@R0   ;SET SECTOR AND CLOCK
11826  050544  042710  000012          3$:   BIC     #MSTCK!MCLK,@R0   ;CLEAR SECTOR AND CLOCK
11827  050550  052710  000002          8$:   BIS     #MCLK,  @R0       ;SET CLOCK
11828  050554  042710  000002                BIC     #MCLK,  @R0       ;CLEAR CLOCK
11829  050560  005302                        DEC     R2                ;BYTE COUNTER
11830  050562  001372                        BNE     8$                ;BRANCH IF BYTE NOT COMPLETE
11831  050564  012702  000007                MOV     #7,     R2        ;SETUP FOR SECOND BYTE
11832  050570  005305                        DEC     R5                ;IS WORD COMPLETE?
11833  050572  001362                        BNE     2$                ;BRANCH IF NOT COMPLETE
11834                                                                    ;TO GIVE SECTOR CLOCK AND CLOCK
11835
11836
11837                                        ;NOW 303 WORDS ARE LEFT AND ALL ARE IDENTICAL
11838
11839  050574  012701  000457                MOV     #303.,  R1        ;WORDS PER SECTOR COUNTER
11840  050600  012705  000002          4$:   MOV     #2,     R5        ;BYTES PER WORD COUNTER
11841  050604  012702  000007          5$:   MOV     #7,     R2        ;BYTE COUNTER (CLOCK COUNTER)
11842  050610  052710  000012                BIS     #MSTCK!MCLK,@R0   ;SET SECTOR CLOCK AND CLOCK
11843  050614  042710  000012                BIC     #MSTCK!MCLK,@R0   ;CLEAR SECTOR CLOCK AND CLOCK
11844  050620  052710  000002          6$:   BIS     #MCLK,  @R0       ;SET CLOCK
```

```
11845  050624  042710  000002          BIC     #MCLK,  @R0     ;RESET CLOCK
11846  050630  005302                  DEC     R2              ;IS BYTE COMPLETE?
11847  050632  001372                  BNE     6$              ;BRANCH IF NOT COMPLETE
11848  050634  005305                  DEC     R5              ;IS WORD COMPLETE?
11849  050636  001362                  BNE     5$              ;BRANCH IF NOT
11850  050640  005301                  DEC     R1              ;IS SECTOR COMPLETE
11851  050642  001356                  BNE     4$              ;BRANCH IF NOT
11852  050644  052710  000010          BIS     #MSTCK,@R0      ;SET SECTOR
11853  050650  042710  000010          BIC     #MSTCK,@R0      ;CLEAR SECTOR
11854  050654  005303                  DEC     R3              ;IS REQUIRED NO OF SECTORS COMPLETE
11855  050656  001317                  BNE     1$              ;BRANCH IF NOT
11856
11857  050660                  7$:
11858  050660  012605                  MOV     (SP)+,R5        ;;POP STACK INTO R5
11859  050662  012604                  MOV     (SP)+,R4        ;;POP STACK INTO R4
11860  050664  012603                  MOV     (SP)+,R3        ;;POP STACK INTO R3
11861  050666  012602                  MOV     (SP)+,R2        ;;POP STACK INTO R2
11862  050670  012601                  MOV     (SP)+,R1        ;;POP STACK INTO R1
11863  050672  012600                  MOV     (SP)+,R0        ;;POP STACK INTO R0
11864  050674  000201                  RTS     R1
11865
11866
11867                   ;;*****************************************************************
11868                   ;READ ONE SECTOR OF DATA
11869                   ;;*****************************************************************
11870
11871  050676  000000          RNO:    0                       ;NO. OF WORDS READ
11872  050700  000000          RCOM:   0                       ;EXTRA STORAGE
11873
11874
11875
11876  050702  012137  050676  REDATA: MOV     (R1)+,@#RNO     ;SAVE NO. OF WORDS ONLY FOR INFORMATION
11877  050706  012137  050700          MOV     (R1)+,@#RCOM    ;EXTRA WORD ONLY FOR INFORMATION
11878  050712  010146                  MOV     R1,-(SP)        ;;PUSH R1 ON STACK
11879  050714  005737  002010          TST     @#TSECC         ;IS THIS AN ECC TEST
11880  050720  001403                  BEQ     1$              ;BRANCH IF NO
11881  050722  012737  177777  042526  MOV     #-1,@#TSECCG    ;THESE BITS ARE TO GENERATE ECC
11882  050730  012702  000402  1$:     MOV     #258.,R2        ;256 WORDS PER SECTOR
11883                                                          ;PLUS 2 ECC WORDS
11884  050734  012703  046330          MOV     #DISK,R3        ;POINTE TO DISK SIMULATION
11885  050740  012337  045322  2$:     MOV     (R3)+,@#WORD    ;READY TO READ CONTENTS
11886  050744  004737  045326          JSR     PC,@#READ               ;READ
11887  050750  005302                  DEC     R2              ;IS 256 WORDS DONE?
11888  050752  001372                  BNE     2$              ;IF NOT BRANCH
11889  050754  005737  002010          TST     @#TSECC         ;IS THIS AN ECC TEST
11890  050760  001012                  BNE     4$              ;BRANCH OUT IF YES
11891  050762  005037  042526          CLR     @#TSECCG        ;NO MORE ECC BITS ARE TO BE GENERATED
11892  050766  012702  000017          MOV     #15.,R2         ;ONE DATA GAP, 14 TOLERANCE GAP
11893  050772  012337  045322  3$:     MOV     (R3)+,@#WORD    ;READY TO READ CONTENTS OF WORD
11894  050776  004737  045326          JSR     PC,@#READ       ;READ
11895  051002  005302                  DEC     R2              ;COUNT
11896  051004  001372                  BNE     3$              ;BRANCH IF 14 NOT DONE
11897  051006                  4$:
11898  051006  012601                  MOV     (SP)+,R1        ;;POP STACK INTO R1
11899  051010  000201                  RTS     R1              ;RETURN
11900
```

```
11901
11902
11903
11904   051012                          RPVECT:
11905   051012  104400  051020                  TYPE    ,65$            ;;TYPE ASCIZ STRING
11906   051016  000411                  BR      64$             ;;GET OVER THE ASCIZ
11907                                   ;;65$:  .ASCIZ  /TRAPED FROM PC = /
11908   051042                          64$:
11909   051042  104401                  TYPOC                           ;TYPE FROM PC
11910   051044  012777  051012  130544        MOV     #RPVECT,@RPVEC          ;RESTORE TRAP TO RP04 VECTOR
11911   051052  000000                  HALT                            ;CHANGE TO CONTINUE
11912
11913
11914
11915
```

```
                                                                                ;****************************
                        .SBTTL  SCOPE HANDLER ROUTINE
                        ;;*****************************************************************
                        ;*THIS ROUTINE CONTROLS THE LOOPING OF SUBTESTS. IT WILL INCREMENT
                        ;*AND LOAD THE TEST NUMBER($TSTNM) INTO THE DISPLAY REG.(DISPLAY<7:0>)
                        ;*AND LOAD THE ERROR FLAG ($ERFLG) INTO DISPLAY<15:08>
                        ;*THE SWITCH OPTIONS PROVIDED BY THIS ROUTINE ARE:
                        ;*SW14=1         LOOP ON TEST
                        ;*SW11=1         INHIBIT ITERATIONS
                        ;*SW09=1         LOOP ON ERROR
                        ;*SW08=1         LOOP ON TEST IN SWR<7:0>
                        ;*CALL
                        ;*      SCOPE           ;;SCOPE=IOT

051054                  $SCOPE:
051054  005037  044526          CLR     @#NOSYNC                ;CLEAR FLAG FOR HEADER ERROR COMMANDS
051060  005037  002010          CLR     @#TSECC                 ;CLEAR FLAG FOR ECC TEST
                                                                ;WHEN =177777 IT IS AN ECC TEST
                                                                ;WHEN =0 IT IS NOT AN ECC TEST

051064  005037  042526          CLR     @#TSECCG                ;EVEN IN AN ECC TEST EVERY CLOCK
                                                                ;IS NOT TO GENERATE ECC
                                                                ;IF =177777 GENERATE ECC
                                                                ;IF =0 DO NOT GENERATE ECC
051070  005037  002012          CLR     @#TESDTE                ;DRIVE TIMING ERROR TEST
051074                  1$:
051074  032777  040000  130036  1$:     BIT     #BIT14,@SWR     ;;LOOP ON PRESENT TEST?
051102  001111                  BNE     $OVER                   ;;YES IF SW14=1
                        ;#####START OF CODE FOR THE XOR TESTER#####
051104  000416          $XTSTR: BR      6$                      ;;IF RUNNING ON THE "XOR" TESTER CHANGE
                                                                ;;THIS INSTRUCTION TO A "NOP" (NOP=240)
051106  013746  000004          MOV     @#ERRVEC,-(SP)          ;;SAVE THE CONTENTS OF THE ERROR VECTOR
051112  012737  051132  000004  MOV     #5$,@#ERRVEC            ;;SET FOR TIMEOUT
051120  005737  177060          TST     @#177060                ;;TIME OUT ON XOR?
051124  012637  000004          MOV     (SP)+,@#ERRVEC          ;;RESTORE THE ERROR VECTOR
051130  000463                  BR      $SVLAD                  ;;GO TO THE NEXT TEST
051132  022626          5$:     CMP     (SP)+,(SP)+             ;;CLEAR THE STACK AFTER A TIME OUT
051134  012637  000004          MOV     (SP)+,@#ERRVEC          ;;RESTORE THE ERROR VECTOR
051140  000423                  BR      7$                      ;;LOOP ON THE PRESENT TEST
051142                  6$:;#####END OF CODE FOR THE XOR TESTER#####
051142  032777  000400  127770  BIT     #BIT08,@SWR             ;;LOOP ON SPEC. TEST?
051150  001404                  BEQ     2$                      ;;BR IF NO
051152  127767  127762  127722  CMPB    @SWR,$TSTNM             ;;ON THE RIGHT TEST?    SWR<7:0>
051160  001462                  BEQ     $OVER                   ;;BR IF YES
051162  105767  127715  2$:     TSTB    $ERFLG                  ;;HAS AN ERROR OCCURRED?
051166  001421                  BEQ     3$                      ;;BR IF NO
051170  126767  127721  127705  CMPB    $ERMAX,$ERFLG           ;;MAX. ERRORS FOR THIS TEST OCCURRED?
051176  101015                  BHI     3$                      ;;BR IF NO
051200  032777  001000  127732  BIT     #BIT09,@SWR             ;;LOOP ON ERROR?
051206  001404                  BEQ     4$                      ;;BR IF NO
051210  016767  127674  127670  7$:     MOV     $LPERR,$LPADR   ;;SET LOOP ADDRESS TO LAST SCOPE
051216  000443                  BR      $OVER
051220  105067  127657  4$:     CLRB    $ERFLG                  ;;ZERO THE ERROR FLAG
051224  005067  127762          CLR     $TIMES                  ;;CLEAR THE NUMBER OF ITERATIONS TO MAKE
051230  000415                  BR      1$                      ;;ESCAPE TO THE NEXT TEST
```

```
11972  051232  032777  004000  127700  3$:     BIT     #BIT11,@SWR     ;;INHIBIT ITERATIONS?
11973  051240  001011                  BNE     1$              ;;BR IF YES
11974  051242  005767  127632          TST     $PASS           ;;IF FIRST PASS OF PROGRAM
11975  051246  001406                  BEQ     1$              ;;      INHIBIT ITERATIONS
11976  051250  005267  127630          INC     $ICNT           ;;INCREMENT ITERATION COUNT
11977  051254  026767  127732  127622  CMP     $TIMES,$ICNT    ;;CHECK THE NUMBER OF ITERATIONS MADE
11978  051262  002021                  BGE     $OVER           ;;BR IF MORE ITERATION REQUIRED
11979  051264  012767  000001  127612  1$:     MOV     #1,$ICNT        ;;REINITIALIZE THE ITERATION COUNTER
11980  051272  016767  000044  127712          MOV     $MXCNT,$TIMES   ;;SET NUMBER OF ITERATIONS TO DO
11981  051300  105267  127576          $SVLAD: INCB    $TSTNM          ;;COUNT TEST NUMBERS
11982  051304  011667  127576                  MOV     (SP),$LPADR     ;;SAVE SCOPE LOOP ADDRESS
11983  051310  011667  127574                  MOV     (SP),$LPERR     ;;SAVE ERROR LOOP ADDRESS
11984  051314  005067  127674                  CLR     $ESCAPE         ;;CLEAR THE ESCAPE FROM ERROR ADDRESS
11985  051320  112767  000001  127567          MOVB    #1,$ERMAX       ;;ONLY ALLOW ONE(1) ERROR ON NEXT TEST
11986  051326  016777  127550  127606  $OVER:  MOV     $TSTNM,@DISPLAY ;;DISPLAY TEST NUMBER
11987  051334  016716  127546                  MOV     $LPADR,(SP)     ;;FUDGE RETURN ADDRESS
11988  051340  000002                          RTI                     ;;FIXES PS
11989  051342  000004          $MXCNT: 4                               ;;MAX. NUMBER OF ITERATIONS
```

```
                                .SBTTL  CONVERT BINARY TO DECIMAL AND TYPE ROUTINE

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

051344                          $TYPDS:
051344  010046                         MOV     R0,-(SP)        ;;PUSH R0 ON STACK
051346  010146                         MOV     R1,-(SP)        ;;PUSH R1 ON STACK
051350  010246                         MOV     R2,-(SP)        ;;PUSH R2 ON STACK
051352  010346                         MOV     R3,-(SP)        ;;PUSH R3 ON STACK
051354  010546                         MOV     R5,-(SP)        ;;PUSH R5 ON STACK
051356  012746  020200                 MOV     #20200,-(SP)    ;;SET BLANK SWITCH AND SIGN
051362  016605  000020                 MOV     20(SP),R5       ;;GET THE INPUT NUMBER
051366  100004                         BPL     1$              ;;BR IF INPUT IS POS.
051370  005405                         NEG     R5              ;;MAKE THE BINARY NUMBER POS.
051372  112766  000055  000001         MOVB    #'-,1(SP)       ;;MAKE THE ASCII NUMBER NEG.
051400  005000                 1$:     CLR     R0              ;;ZERO THE CONSTANTS INDEX
051402  012703  051560                 MOV     #$DBLK,R3       ;;SETUP THE OUTPUT POINTER
051406  112723  000040                 MOVB    #' ,(R3)+       ;;SET THE FIRST CHARACTER TO A BLANK
051412  005002                 2$:     CLR     R2              ;;CLEAR THE BCD NUMBER
051414  016001  051550                 MOV     $DTBL(R0),R1    ;;GET THE CONSTANT
051420  160105                 3$:     SUB     R1,R5           ;;FORM THIS BCD DIGIT
051422  002402                         BLT     4$              ;;BR IF DONE
051424  005202                         INC     R2              ;;INCREASE THE BCD DIGIT BY 1
051426  000774                         BR      3$
051430  060105                 4$:     ADD     R1,R5           ;;ADD BACK THE CONSTANT
051432  005702                         TST     R2              ;;CHECK IF BCD DIGIT=0
051434  001002                         BNE     5$              ;;FALL THROUGH IF 0
051436  105716                         TSTB    (SP)            ;;STILL DOING LEADING 0'S?
051440  100407                         BMI     7$              ;;BR IF YES
051442  106316                 5$:     ASLB    (SP)            ;;MSD?
051444  103003                         BCC     6$              ;;BR IF NO
051446  116663  000001  177777         MOVB    1(SP),-1(R3)    ;;YES--SET THE SIGN
051454  052702  000060         6$:     BIS     #'0,R2          ;;MAKE THE BCD DIGIT ASCII
051460  052702  000040         7$:     BIS     #' ,R2          ;;MAKE IT A SPACE IF NOT ALREADY A DIGIT
051464  110223                         MOVB    R2,(R3)+        ;;PUT THIS CHARACTER IN THE OUTPUT BUFFER
051466  005720                         TST     (R0)+           ;;JUST INCREMENTING
051470  020027  000010                 CMP     R0,#10          ;;CHECK THE TABLE INDEX
051474  002746                         BLT     2$              ;;GO DO THE NEXT DIGIT
051476  003002                         BGT     8$              ;;GO TO EXIT
051500  010502                         MOV     R5,R2           ;;GET THE LSD
051502  000764                         BR      6$              ;;GO CHANGE TO ASCII
051504  105726                 8$:     TSTB    (SP)+           ;;WAS THE LSD THE FIRST NON-ZERO?
051506  100003                         BPL     9$              ;;BR IF NO
051510  116663  177777  177776         MOVB    -1(SP),-2(R3)   ;;YES--SET THE SIGN FOR TYPING
051516  105013                 9$:     CLRB    (R3)            ;;SET THE TERMINATOR
051520  012605                         MOV     (SP)+,R5        ;;POP STACK INTO R5
051522  012603                         MOV     (SP)+,R3        ;;POP STACK INTO R3
051524  012602                         MOV     (SP)+,R2        ;;POP STACK INTO R2
```

```
12046  051526  012601                          MOV     (SP)+,R1        ;;POP STACK INTO R1
12047  051530  012600                          MOV     (SP)+,R0        ;;POP STACK INTO R0
12048  051532  104400  051560                  TYPE    $DBLK           ;;NOW TYPE THE NUMBER
12049  051536  016666  000002  000004          MOV     2(SP),4(SP)     ;;ADJUST THE STACK
12050  051544  012616                          MOV     (SP)+,(SP)
12051  051546  000002                          RTI                     ;;RETURN TO USER
12052  051550  023420                $DTBL:    10000.
12053  051552  001750                          1000.
12054  051554  000144                          100.
12055  051556  000012                          10.
12056  051560  000004                $DBLK:    .BLKW   4
12057                                 .SBTTL  TYPE ROUTINE
12058
12059                                 ;;*****************************************************************
12060                                 ;*ROUTINE TO TYPE ASCIZ MESSAGE. MESSAGE MUST TERMINATE WITH A 0 BYTE.
12061                                 ;*THE ROUTINE WILL INSERT A NUMBER OF NULL CHARACTERS AFTER A LINE FEED.
12062                                 ;*NOTE1:        $NULL CONTAINS THE CHARACTER TO BE USED AS THE FILLER CHARACTER.
12063                                 ;*NOTE2:        $FILLS CONTAINS THE NUMBER OF FILLER CHARACTERS REQUIRED.
12064                                 ;*NOTE3:        $FILLC CONTAINS THE CHARACTER TO FILL AFTER.
12065                                 ;*
12066                                 ;*CALL:
12067                                 ;*1) USING A TRAP INSTRUCTION
12068                                 ;*      TYPE    ,MESADR         ;;MESADR IS FIRST ADDRESS OF AN ASCIZ STRING
12069                                 ;*OR
12070                                 ;*      TYPE
12071                                 ;*      MESADR
12072                                 ;*
12073
12074  051570  105767  127363  $TYPE:  TSTB    $TPFLG          ;;IS THERE A TERMINAL?
12075  051574  100002                 BPL     1$              ;;BR IF YES
12076  051576  000000                 HALT                    ;;HALT HERE IF NO TERMINAL
12077  051600  000407                 BR      3$              ;;LEAVE
12078  051602  010046          1$:    MOV     R0,-(SP)        ;;SAVE R0
12079  051604  017600  000002        MOV     @2(SP),R0       ;;GET ADDRESS OF ASCIZ STRING
12080  051610  112046          2$:    MOVB    (R0)+,-(SP)     ;;PUSH CHARACTER TO BE TYPED ONTO STACK
12081  051612  001005                 BNE     4$              ;;BR IF IT ISN'T THE TERMINATOR
12082  051614  005726                 TST     (SP)+           ;;IF TERMINATOR POP IT OFF THE STACK
12083  051616  012600          60$:   MOV     (SP)+,R0        ;;RESTORE R0
12084  051620  062716  000002  3$:    ADD     #2,(SP)         ;;ADJUST RETURN PC
12085  051624  000002                 RTI                     ;;RETURN
12086  051626  122716  000011  4$:    CMPB    #HT,(SP)        ;;BRANCH IF <HT>
12087  051632  001430                 BEQ     8$
12088  051634  122716  000200        CMPB    #CRLF,(SP)      ;;BRANCH IF NOT <CRLF>
12089  051640  001006                 BNE     5$
12090  051642  005726                 TST     (SP)+           ;;POP  <CR><LF> EQUIV
12091  051644  104400                 TYPE                    ;;TYPE A CR AND LF
12092  051646  001223                 $CRLF
12093  051650  105067  000130        CLRB    $CHARCNT        ;;CLEAR CHARACTER COUNT
12094  051654  000755                 BR      2$              ;;GET NEXT CHARACTER
12095  051656  004767  000056  5$:    JSR     PC,$TYPEC       ;;GO TYPE THIS CHARACTER
12096  051662  126726  127270  6$:    CMPB    $FILLC,(SP)+    ;;IS IT TIME FOR FILLER CHARS.?
12097  051666  001350                 BNE     2$              ;;IF NO GO GET NEXT CHAR.
12098  051670  016746  127260        MOV     $NULL,-(SP)     ;;GET # OF FILLER CHARS. NEEDED
12099                                                         ;;AND THE NULL CHAR.
12100  051674  105366  000001  7$:    DECB    1(SP)           ;;DOES A NULL NEED TO BE TYPED?
12101  051700  002770                 BLT     6$              ;;BR IF NO--GO POP THE NULL OFF OF STACK
```

```
12102  051702  004767  000032            JSR     PC,$TYPEC       ;;GO TYPE A NULL
12103  051706  105367  000072            DECB    $CHARCNT        ;;DO NOT COUNT AS A COUNT
12104  051712  000770                      BR      7$              ;;LOOP
12105
12106                                  ;HORIZONTAL TAB PROCESSOR
12107
12108  051714  112716  000040          8$:     MOVB    #' ,(SP)        ;;REPLACE TAB WITH SPACE
12109  051720  004767  000014          9$:     JSR     PC,$TYPEC       ;;TYPE A SPACE
12110  051724  132767  000007  000052          BITB    #7,$CHARCNT     ;;BRANCH IF NOT AT
12111  051732  001372                          BNE     9$              ;;TAB STOP
12112  051734  005726                          TST     (SP)+           ;;POP SPACE OFF STACK
12113  051736  000724                          BR      2$              ;;GET NEXT CHARACTER
12114  051740  105777  127204        $TYPEC: TSTB    @$TPS           ;;WAIT UNTIL PRINTER IS READY
12115  051744  100375                          BPL     $TYPEC
12116  051746  116677  000002  127176        MOVB    2(SP),@$TPB     ;;LOAD CHAR TO BE TYPED INTO DATA REG.
12117  051754  122766  000015  000002        CMPB    #CR,2(SP)       ;;IS CHARACTER A CARRIAGE RETURN?
12118  051762  001003                          BNE     1$              ;;BRANCH IF NO
12119  051764  105067  000014                CLRB    $CHARCNT        ;;YES--CLEAR CHARACTER COUNT
12120  051770  000406                          BR      $TYPEX          ;;EXIT
12121  051772  122766  000012  000002  1$:     CMPB    #LF,2(SP)       ;;IS CHARACTER A LINE FEED?
12122  052000  001402                          BEQ     $TYPEX          ;;BRANCH IF YES
12123  052002  105227                          INCB    (PC)+           ;;COUNT THE CHARACTER
12124  052004  000000                $CHARCNT:.WORD  0               ;;CHARACTER COUNT STORAGE
12125  052006  000207                $TYPEX: RTS     PC
12126
12127                                  .SBTTL  TTY INPUT ROUTINE
12128
12129                                  ;;**************************************************************
12130                                  .ENABL  LSB
12131  052010  000000                $TKCNT: .WORD   0               ;;NUMBER OF ITEMS IN QUEUE
12132  052012  000000                $TKQIN: .WORD   0               ;;INPUT POINTER
12133  052014  000000                $TKQOUT: .WORD  0               ;;OUTPUT POINTER
12134  052016  000011                $TKQSRT: .BLKB  9.              ;;TTY KEYBOARD QUEUE
12135           052027                $TKQEND=.
12136           052030                .EVEN
12137
12138                                  ;*TK INITIALIZE ROUTINE
12139                                  ;*THIS ROUTINE WILL INITIALIZE THE TTY KEYBOARD INPUT QUEUE
12140                                  ;*SETUP THE INTERRUPT VECTOR AND TURN ON THE KEYBOARD INTERRUPT
12141                                  ;
12142                                  ;*CALL:
12143                                  ;*      JSR     PC,$TKINT
12144                                  ;*      RETURN
12145                                  ;
12146  052030  005067  177754        $TKINT: CLR     $TKCNT          ;;CLEAR COUNT OF ITEMS IN QUEUE
12147  052034  012767  052016  177750        MOV     #$TKQSRT,$TKQIN ;;MOVE THE STARTING ADDRESS OF THE
12148  052042  016767  177744  177744        MOV     $TKQIN,$TKQOUT  ;;QUEUE INTO THE INPUT & OUTPUT POINTERS.
12149  052050  012737  052100  000060        MOV     #$TKSRV,@#TKVEC ;;INITIALIZE THE KEYBOARD VECTOR
12150  052056  012737  000200  000062        MOV     #200,@#TKVEC+2  ;;"BR" LEVEL 4
12151  052064  005777  127056                TST     @$TKB           ;;CLEAR DONE FLAG
12152  052070  012777  000100  127046        MOV     #100,@$TKS      ;;ENABLE TTY KEYBOARD INTERRUPT
12153  052076  000207                        RTS     PC              ;;RETURN TO CALLER
12154
12155                                  ;*TK SERVICE ROUTINE
12156                                  ;*THIS ROUTINE WILL SERVICE THE TTY KEYBOARD INTERRUPT
12157                                  ;*BY READING THE CHARACTER FROM THE INPUT BUFFER AND PUTTING
```

```
12158                                   ;*IT IN THE QUEUE.
12159                                   ;*IF THE CHARACTER IS A "CONTROL-C" (↑C) $TKINT IS CALLED AND
12160                                   ;*UPON RETURN EXIT IS MADE TO THE "CONTROL-C" RESTART ADDRESS (OPERSEL)
12161                                   ;
12162  052100  117746  127042          $TKSRV: MOVB    @$TKB,-(SP)     ;;PICKUP THE CHARACTER
12163  052104  042716  177600                  BIC     #↑C177,(SP)     ;;STRIP THE JUNK
12164  052110  021627  000003                  CMP     (SP),#3         ;;IS IT A CONTROL C?
12165  052114  001007                          BNE     1$              ;;BRANCH IF NO
12166  052116  104400  052507                  TYPE    ,$CNTLC         ;;TYPE A CONTROL-C (↑C)
12167  052122  004767  177702                  JSR     PC,$TKINT       ;;INIT THE KEYBOARD
12168  052126  005726                          TST     (SP)+           ;;CLEAN UP STACK
12169  052130  000167  164554                  JMP     OPERSEL         ;;CONTROL C RESTART
12170
12171  052134                          1$:
12172  052134  022767  000011  177646          CMP     #9.,$TKCNT      ;;IS THE QUEUE FULL?
12173  052142  001004                          BNE     3$              ;;BRANCH IF NO
12174  052144  104400  001216                  TYPE    ,$BELL          ;;RING THE TTY BELL
12175  052150  005726                          TST     (SP)+           ;;CLEAN CHARACTER OFF OF STACK
12176  052152  000451                          BR      5$              ;;EXIT
12177  052154  021627  000023          3$:     CMP     (SP),#23        ;;IS IT A CONTROL-S?
12178  052160  001021                          BNE     32$             ;;BRANCH IF NO
12179  052162  005077  126756                  CLR     @$TKS           ;;DISABLE TTY KEYBOARD INTERRUPTS
12180  052166  005726                          TST     (SP)+           ;;CLEAN CHAR OFF STACK
12181  052170  105777  126750          31$:    TSTB    @$TKS           ;;WAIT FOR A CHAR
12182  052174  100375                          BPL     31$             ;;LOOP UNTIL ITS THERE
12183  052176  117746  126744                  MOVB    @$TKB,-(SP)     ;;GET THE CHARACTER
12184  052202  042716  177600                  BIC     #↑C177,(SP)     ;;MAKE IT 7-BIT ASCII
12185  052206  022627  000021                  CMP     (SP)+,#21       ;;IS IT A CONTROL-Q?
12186  052212  001366                          BNE     31$             ;;BRANCH IF NO
12187  052214  012777  000100  126722          MOV     #100,@$TKS      ;;REENABLE TTY KEYBOARD INTERRUPTS
12188  052222  000002                          RTI                     ;;RETURN
12189  052224  005267  177560          32$:    INC     $TKCNT          ;;COUNT THIS CHARACTER
12190  052230  021627  000140                  CMP     (SP),#140       ;;IS IT UPPER CASE?
12191  052234  002405                          BLT     4$              ;;BRANCH IF YES
12192  052236  021627  000175                  CMP     (SP),#175       ;;IS IT A SPECIAL CHAR?
12193  052242  003002                          BGT     4$              ;;BRANCH IF YES
12194  052244  042716  000040                  BIC     #40,(SP)        ;;MAKE IT UPPER CASE
12195  052250  112677  177536          4$:     MOVB    (SP)+,@$TKQIN   ;;AND PUT IT IN QUEUE
12196  052254  005267  177532                  INC     $TKQIN          ;;UPDATE THE POINTER
12197  052260  026727  177526  052027          CMP     $TKQIN,#$TKQEND ;;GO OFF THE END?
12198  052266  001003                          BNE     5$              ;;BRANCH IF NO
12199  052270  012767  052016  177514          MOV     #$TKQSRT,$TKQIN ;;RESET THE POINTER
12200  052276  000002                  5$:     RTI                     ;;RETURN
12201
12202                                   .DSABL  LSB
12203
12204
12205                                   ;;*****************************************************************
12206                                   ;*THIS ROUTINE WILL INPUT A SINGLE CHARACTER FROM THE TTY
12207                                   ;*CALL:
12208                                   ;*      RDCHR                   ;;GET A CHARACTER FROM THE QUEUE
12209                                   ;*      RETURN HERE             ;;CHARACTER IS ON THE STACK
12210                                   ;*                              ;;WITH PARITY BIT STRIPPED OFF
12211                                   ;
12212
12213  052300  011646                  $RDCHR: MOV     (SP),-(SP)      ;;PUSH DOWN THE PC AND
```

```
12214  052302  016666  000004  000002          MOV     4(SP),2(SP)     ;;THE PS
12215  052310  005066  000004                  CLR     4(SP)           ;;GET READY FOR A CHARACTER
12216  052314  005046                          CLR     -(SP)           ;;PUT NEW PS ON STACK
12217  052316  012746  052324                  MOV     #64$,-(SP)      ;;PUT NEW PC ON  STACK
12218  052322  000002                          RTI                     ;;POP NEW PC AND PS
12219  052324                          64$:
12220  052324  005767  177460          1$:     TST     $TKCNT          ;;WAIT ON A CHARACTER
12221  052330  001775                          BEQ     1$
12222  052332  005367  177452                  DEC     $TKCNT          ;;DECREMENT THE COUNTER
12223  052336  117766  177452  000004          MOVB    @$TKQOUT,4(SP)  ;;GET ONE CHARACTER
12224  052344  005267  177444                  INC     $TKQOUT         ;;UPDATE THE POINTER
12225  052350  026727  177440  052027          CMP     $TKQOUT,#$TKQEND ;;DID IT GO OFF OF THE END?
12226  052356  001003                          BNE     2$              ;;BRANCH IF NO
12227  052360  012767  052016  177426          MOV     #$TKQSRT,$TKQOUT ;;RESET THE POINTER
12228  052366  000002                  2$:     RTI                     ;;RETURN
12229                                  ;;**************************************************************
12230                                  ;*THIS ROUTINE WILL INPUT A STRING FROM THE TTY
12231                                  ;*CALL:
12232                                  ;*      RDLIN                   ;;INPUT A STRING FROM THE TTY
12233                                  ;*      RETURN HERE             ;;ADDRESS OF FIRST CHARACTER WILL BE ON THE STACK
12234                                  ;*                              ;;TERMINATOR WILL BE A BYTE OF ALL 0'S
12235
12236  052370  010346                  $RDLIN: MOV     R3,-(SP)        ;;SAVE R3
12237  052372  012703  052476          1$:     MOV     #$TTYIN,R3      ;;GET ADDRESS
12238  052376  022703  052507          2$:     CMP     #$TTYIN+9.,R3   ;;BUFFER FULL?
12239  052402  101405                          BLOS    4$              ;;BR IF YES
12240  052404  104405                          RDCHR                   ;;GO READ ONE CHARACTER FROM THE TTY
12241  052406  112613                          MOVB    (SP)+,(R3)      ;;GET CHARACTER
12242  052410  122713  000177          10$:    CMPB    #177,(R3)       ;;IS IT A RUBOUT
12243  052414  001003                          BNE     3$              ;;SKIP IF NOT
12244  052416  104400  001222          4$:     TYPE    ,$QUES          ;;TYPE A '?'
12245  052422  000763                          BR      1$              ;;CLEAR THE BUFFER AND LOOP
12246  052424  111367  000044          3$:     MOVB    (R3),9$         ;;ECHO THE CHARACTER
12247  052430  104400  052474                  TYPE    ,9$
12248  052434  122723  000015                  CMPB    #15,(R3)+       ;;CHECK FOR RETURN
12249  052440  001356                          BNE     2$              ;;LOOP IF NOT RETURN
12250  052442  105063  177777                  CLRB    -1(R3)          ;;CLEAR RETURN (THE 15)
12251  052446  104400  001224                  TYPE    ,$LF            ;;TYPE A LINE FEED
12252  052452  012603                          MOV     (SP)+,R3        ;;RESTORE R3
12253  052454  011646                          MOV     (SP),-(SP)      ;;ADJUST THE STACK AND PUT ADDRESS OF THE
12254  052456  016666  000004  000002          MOV     4(SP),2(SP)     ;;      FIRST ASCII CHARACTER ON IT
12255  052464  012766  052476  000004          MOV     #$TTYIN,4(SP)
12256  052472  000002                          RTI                     ;;RETURN
12257  052474     000                  9$:     .BYTE   0               ;;STORAGE FOR ASCII CHAR. TO TYPE
12258  052475     000                          .BYTE   0               ;;TERMINATOR
12259  052476  000011                  $TTYIN: .BLKB   9.              ;;RESERVE 9. BYTES FOR TTY INPUT
12260  052507     136  006503  000012  $CNTLC: .ASCIZ  /↑C/<15><12>    ;;CONTROL "C"
12261  052514  052536  005015     000  $CNTLU: .ASCIZ  /↑U/<15><12>    ;;CONTROL "U"
12262  052521     136  006507  000012  $CNTLG: .ASCIZ  /↑G/<15><12>    ;;CONTROL "G"
12263  052526  005015  053523  020122  $MSWR:  .ASCIZ  <15><12>/SWR = /
12264  052534  020075     000
12265  052537     040  047040  053505  $MNEW:  .ASCIZ  /  NEW = /
12266  052544  036440  000040
12267                                                           ;FROM THE TTY
12268
12269
```

```
12270
12271                                   .SBTTL  READ AN OCTAL NUMBER FROM THE TTY
12272
12273                                   ;;**************************************************************
12274                                   ;*THIS ROUTINE WILL READ AN OCTAL (ASCII) NUMBER FROM THE TTY AND
12275                                   ;*CHANGE IT TO BINARY.
12276                                   ;*THE INPUT CHARACTERS WILL BE CHECKED TO INSURED THEY ARE LEGAL
12277                                   ;*OCTAL DIGITS. IF AN ILLEGAL CHARACTER IS READ A "?" WILL BE TYPED
12278                                   ;*FOLLOWED BY A CARRIAGE RETURN-LINE FEED. THE COMPLETE NUMBER MUST
12279                                   ;*THEN BE RETYPED. THE INPUT IS TERMINATED BY TYPING A CARRIAGE RETURN.
12280                                   ;*CALL:
12281                                   ;*      RDOCT                   ;;READ AN OCTAL NUMBER
12282                                   ;*      RETURN HERE             ;;LOW ORDER BITS ARE ON TOP OF THE STACK
12283                                   ;*                              ;;HIGH ORDER BITS ARE IN $HIOCT
12284
12285  052550  011646                $RDOCT: MOV     (SP),-(SP)      ;;PROVIDE SPACE FOR THE
12286  052552  016666  000004  000002        MOV     4(SP),2(SP)     ;;INPUT NUMBER
12287  052560  010046                        MOV     R0,-(SP)        ;;PUSH R0 ON STACK
12288  052562  010146                        MOV     R1,-(SP)        ;;PUSH R1 ON STACK
12289  052564  010246                        MOV     R2,-(SP)        ;;PUSH R2 ON STACK
12290  052566  104406                1$:     RDLIN                   ;;READ AN ASCIZ LINE
12291  052570  012600                        MOV     (SP)+,R0        ;;GET ADDRESS OF 1ST CHARACTER
12292  052572  010067  000100               MOV     R0,5$           ;;AND SAVE IT
12293  052576  005001                        CLR     R1              ;;CLEAR DATA WORD
12294  052600  005002                        CLR     R2
12295  052602  112046                2$:     MOVB    (R0)+,-(SP)     ;;PICKUP THIS CHARACTER
12296  052604  001420                        BEQ     3$              ;;IF ZERO GET OUT
12297  052606  122716  000060               CMPB    #'0,(SP)        ;;MAKE SURE THIS CHARACTER
12298  052612  003026                        BGT     4$              ;;IS AN OCTAL DIGIT
12299  052614  122716  000067               CMPB    #'7,(SP)
12300  052620  002423                        BLT     4$
12301  052622  006301                        ASL     R1              ;;*2
12302  052624  006102                        ROL     R2
12303  052626  006301                        ASL     R1              ;;*4
12304  052630  006102                        ROL     R2
12305  052632  006301                        ASL     R1              ;;*8
12306  052634  006102                        ROL     R2
12307  052636  042716  177770               BIC     #^C7,(SP)       ;;STRIP THE ASCII JUNK
12308  052642  062601                        ADD     (SP)+,R1        ;;ADD IN THIS DIGIT
12309  052644  000756                        BR      2$              ;;LOOP
12310  052646  005726                3$:     TST     (SP)+           ;;CLEAN TERMINATOR FROM STACK
12311  052650  010166  000012               MOV     R1,12(SP)       ;;SAVE THE RESULT
12312  052654  010267  000026               MOV     R2,$HIOCT
12313  052660  012602                        MOV     (SP)+,R2        ;;POP STACK INTO R2
12314  052662  012601                        MOV     (SP)+,R1        ;;POP STACK INTO R1
12315  052664  012600                        MOV     (SP)+,R0        ;;POP STACK INTO R0
12316  052666  000002                        RTI                     ;;RETURN
12317  052670  005726                4$:     TST     (SP)+           ;;CLEAN PARTIAL FROM STACK
12318  052672  105010                        CLRB    (R0)            ;;SET A TERMINATOR
12319  052674  104400                        TYPE                    ;;TYPE UP THRU THE BAD CHAR.
12320  052676  000000                5$:     .WORD   0
12321  052700  104400  001222               TYPE    ,$QUES          ;;"?" "CR" & "LF"
12322  052704  000730                        BR      1$              ;;TRY AGAIN
12323  052706  000000                $HIOCT: .WORD   0               ;;HIGH ORDER BITS GO HERE
```

```
                                .SBTTL  ERROR HANDLER ROUTINE

                                ;;**********************************************************************
                                ;*THIS ROUTINE WILL INCREMENT THE ERROR FLAG AND THE ERROR COUNT,
                                ;*SAVE THE ERROR ITEM NUMBER AND THE ADDRESS OF THE ERROR CALL
                                ;*AND GO TO $ERRTYP ON ERROR
                                ;*THE SWITCH OPTIONS PROVIDED BY THIS ROUTINE ARE:
                                ;*SW15=1         HALT ON ERROR
                                ;*SW13=1         INHIBIT ERROR TYPEOUTS
                                ;*SW10=1         BELL ON ERROR
                                ;*SW09=1         LOOP ON ERROR
                                ;*CALL
                                ;*      ERROR   N       ;;ERROR=EMT AND N=ERROR ITEM NUMBER

052710                          $ERROR:

052710  012737  177777  001772          MOV     #-1,@#ERFLG$    ;SET ERROR FLAG
052716                          REGSA1:

                                7$:     INCB    $ERFLG          ;;SET THE ERROR FLAG
052716  105267  126161                  BEQ     7$              ;;DON'T LET THE FLAG GO TO ZERO
052722  001775
052724  016777  126152  126210         MOV     $TSTNM,@DISPLAY ;;DISPLAY TEST NUMBER AND ERROR FLAG
052732  032777  002000  126200         BIT     #BIT10,@SWR     ;;BELL ON ERROR?
052740  001402                  BEQ     1$              ;;NO - SKIP
052742  104400  001216          TYPE    ,$BELL          ;;RING BELL
052746  005267  126140          1$:     INC     $ERTTL          ;;COUNT THE NUMBER OF ERRORS
052752  011667  126140          MOV     (SP),$ERRPC     ;;GET ADDRESS OF ERROR INSTRUCTION
052756  162767  000002  126132  SUB     #2,$ERRPC
052764  117767  126126  126122  MOVB    @$ERRPC,$ITEMB  ;;STRIP AND SAVE THE ERROR ITEM CODE
052772  032777  020000  126140  BIT     #BIT13,@SWR     ;;SKIP TYPEOUT IF SET
053000  001004                  BNE     20$             ;;SKIP TYPEOUTS
053002  004767  000044          JSR     PC,$ERRTYP      ;;GO TO USER ERROR ROUTINE
053006  104400  001223          TYPE    ,$CRLF
053012                          20$:
053012  005777  126122          2$:     TST     @SWR            ;;HALT ON ERROR
053016  100001                  BPL     3$              ;;SKIP IF CONTINUE
053020  000000                  HALT                    ;;HALT ON ERROR!
053022  032777  001000  126110  3$:     BIT     #BIT09,@SWR     ;;LOOP ON ERROR SWITCH SET?
053030  001402                  BEQ     4$              ;;BR IF NO
053032  016716  126052          MOV     $LPERR,(SP)     ;;FUDGE RETURN FOR LOOPING
053036  005767  126152          4$:     TST     $ESCAPE         ;;CHECK FOR AN ESCAPE ADDRESS
053042  001402                  BEQ     5$              ;;BR IF NONE
053044  016716  126144          MOV     $ESCAPE,(SP)    ;;FUDGE RETURN ADDRESS FOR ESCAPE
053050                          5$:
053050  000002                  RTI                     ;;RETURN
                                .SBTTL  ERROR MESSAGE TYPEOUT ROUTINE

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

053052                          $ERRTYP:
053052  104400  001223                  TYPE    ,$CRLF          ;;"CARRIAGE RETURN" & "LINE FEED"
053056  010046                          MOV     R0,-(SP)        ;;SAVE R0
```

```
12380  053060  005000                    CLR     R0              ;;PICKUP THE ITEM INDEX
12381  053062  153700  001114            BISB    @#$ITEMB,R0
12382  053066  001004                    BNE     1$              ;;IF ITEM NUMBER IS ZERO, JUST
12383                                                            ;;TYPE THE PC OF THE ERROR
12384  053070  016746  126022            MOV     $ERRPC,-(SP)    ;;SAVE $ERRPC FOR TYPEOUT
12385                                                            ;;ERROR ADDRESS
12386  053074  104401                    TYPOC                   ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
12387  053076  000445                    BR      10$             ;;GET OUT
12388  053100  005300            1$:     DEC     R0              ;;ADJUST THE INDEX SO THAT IT WILL
12389  053102  006300                    ASL     R0              ;;      WORK FOR THE ERROR TABLE
12390  053104  006300                    ASL     R0
12391  053106  006300                    ASL     R0
12392  053110  062700  001226            ADD     #$ERRTB,R0      ;;FORM TABLE POINTER
12393  053114  012067  000004            MOV     (R0)+,2$        ;;PICKUP "ERROR MESSAGE" POINTER
12394  053120  001404                    BEQ     3$              ;;SKIP TYPEOUT IF NO POINTER
12395  053122  104400                    TYPE                    ;;TYPE THE "ERROR MESSAGE"
12396  053124  000000            2$:     .WORD   0               ;;"ERROR MESSAGE" POINTER GOES HERE
12397  053126  104400  001223            TYPE    ,$CRLF          ;;"CARRIAGE RETURN" & "LINE FEED"
12398  053132  012067  000004    3$:     MOV     (R0)+,4$        ;;PICKUP "DATA HEADER" POINTER
12399  053136  001404                    BEQ     5$              ;;SKIP TYPEOUT IF 0
12400  053140  104400                    TYPE                    ;;TYPE THE "DATA HEADER"
12401  053142  000000            4$:     .WORD   0               ;;"DATA HEADER" POINTER GOES HERE
12402  053144  104400  001223            TYPE    ,$CRLF          ;;"CARRIAGE RETURN" & "LINE FEED"
12403  053150  010146            5$:     MOV     R1,-(SP)        ;;SAVE R1
12404  053152  012001                    MOV     (R0)+,R1        ;;PICKUP "DATA TABLE" POINTER
12405  053154  001415                    BEQ     9$              ;;BR IF NO DATA TO BE TYPED
12406  053156  012000                    MOV     (R0)+,R0        ;;PICKUP "DATA FORMAT" POINTER
12407  053160  105720            6$:     TSTB    (R0)+           ;;"OCTAL" OR "DECIMAL"
12408  053162  001003                    BNE     7$              ;;BR IF DECIMAL
12409  053164  013146                    MOV     @(R1)+,-(SP)    ;;SAVE @(R1)+ FOR TYPEOUT
12410  053166  104401                    TYPOC                   ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
12411  053170  000402                    BR      8$
12412  053172                    7$:
12413  053172  013146                    MOV     @(R1)+,-(SP)    ;;SAVE @(R1)+ FOR TYPEOUT
12414  053174  104404                    TYPDS                   ;;GO TYPE--DECIMAL ASCII WITH SIGN
12415  053176  005711            8$:     TST     (R1)            ;;IS THERE ANOTHER NUMBER?
12416  053200  001403                    BEQ     9$              ;;BR IF NO
12417  053202  104400  053222            TYPE    ,11$            ;;TYPE TWO(2) SPACES
12418  053206  000764                    BR      6$              ;;LOOP
12419
12420  053210  012601            9$:     MOV     (SP)+,R1        ;;RESTORE R1
12421  053212  012600            10$:    MOV     (SP)+,R0        ;;RESTORE R0
12422  053214  104400  001223            TYPE    ,$CRLF          ;;"CARRIAGE RETURN" & "LINE FEED"
12423  053220  000207                    RTS     PC              ;;RETURN
12424  053222  020040      000   11$:    .ASCIZ  /  /            ;;TWO(2) SPACES
12425          053226                    .EVEN
12426                            ;*****************************************************************
12427                            .SBTTL  BINARY TO OCTAL (ASCII) AND TYPE
12428
12429                            ;;*****************************************************************
12430                            ;*THIS ROUTINE IS USED TO CHANGE A 16-BIT BINARY NUMBER TO A 6-DIGIT
12431                            ;*OCTAL (ASCII) NUMBER AND TYPE IT.
12432                            ;*$TYPOS---ENTER HERE TO SETUP SUPPRESS ZEROS AND NUMBER OF DIGITS TO TYPE
12433                            ;*CALL:
12434                            ;*      MOV     NUM,-(SP)       ;;NUMBER TO BE TYPED
12435                            ;*      TYPOS                   ;;CALL FOR TYPEOUT
```

```
12436                                   ;*      .BYTE   N               ;;N=1 TO 6 FOR NUMBER OF DIGITS TO TYPE
12437                                   ;*      .BYTE   M               ;;M=1 OR 0
12438                                   ;*                              ;;1=TYPE LEADING ZEROS
12439                                   ;*                              ;;0=SUPPRESS LEADING ZEROS
12440                                   ;*
12441                                   ;*$TYPON----ENTER HERE TO TYPE OUT WITH THE SAME PARAMETERS AS THE LAST
12442                                   ;*$TYPOS OR $TYPOC
12443                                   ;*CALL:
12444                                   ;*      MOV     NUM,-(SP)       ;;NUMBER TO BE TYPED
12445                                   ;*      TYPON                   ;;CALL FOR TYPEOUT
12446                                   ;*
12447                                   ;*$TYPOC---ENTER HERE FOR TYPEOUT OF A 16 BIT NUMBER
12448                                   ;*CALL:
12449                                   ;*      MOV     NUM,-(SP)       ;;NUMBER TO BE TYPED
12450                                   ;*      TYPOC                   ;;CALL FOR TYPEOUT
12451
12452  053226  017646  000000          $TYPOS: MOV     @(SP),-(SP)     ;;PICKUP THE MODE
12453  053232  116667  000001  000211          MOVB    1(SP),$OFILL    ;;LOAD ZERO FILL SWITCH
12454  053240  112667  000207                  MOVB    (SP)+,$OMODE+1  ;;NUMBER OF DIGITS TO TYPE
12455  053244  062716  000002                  ADD     #2,(SP)         ;;ADJUST RETURN ADDRESS
12456  053250  000406                          BR      $TYPON
12457  053252  112767  000001  000171  $TYPOC: MOVB    #1,$OFILL       ;;SET THE ZERO FILL SWITCH
12458  053260  112767  000006  000165          MOVB    #6,$OMODE+1     ;;SET FOR SIX(6) DIGITS
12459  053266  112767  000005  000154  $TYPON: MOVB    #5,$OCNT        ;;SET THE ITERATION COUNT
12460  053274  010346                          MOV     R3,-(SP)        ;;SAVE R3
12461  053276  010446                          MOV     R4,-(SP)        ;;SAVE R4
12462  053300  010546                          MOV     R5,-(SP)        ;;SAVE R5
12463  053302  116704  000145                  MOVB    $OMODE+1,R4     ;;GET THE NUMBER OF DIGITS TO TYPE
12464  053306  005404                          NEG     R4
12465  053310  062704  000006                  ADD     #6,R4           ;;SUBTRACT IT FOR MAX. ALLOWED
12466  053314  110467  000132                  MOVB    R4,$OMODE       ;;SAVE IT FOR USE
12467  053320  116704  000125                  MOVB    $OFILL,R4       ;;GET THE ZERO FILL SWITCH
12468  053324  016605  000012                  MOV     12(SP),R5       ;;PICKUP THE INPUT NUMBER
12469  053330  005003                          CLR     R3              ;;CLEAR THE OUTPUT WORD
12470  053332  006105                  1$:     ROL     R5              ;;ROTATE MSB INTO "C"
12471  053334  000404                          BR      3$              ;;GO DO MSB
12472  053336  006105                  2$:     ROL     R5              ;;FORM THIS DIGIT
12473  053340  006105                          ROL     R5
12474  053342  006105                          ROL     R5
12475  053344  010503                          MOV     R5,R3
12476  053346  006103                  3$:     ROL     R3              ;;GET LSB OF THIS DIGIT
12477  053350  105367  000076                  DECB    $OMODE          ;;TYPE THIS DIGIT?
12478  053354  100016                          BPL     7$              ;;BR IF NO
12479  053356  042703  177770                  BIC     #177770,R3      ;;GET RID OF JUNK
12480  053362  001002                          BNE     4$              ;;TEST FOR 0
12481  053364  005704                          TST     R4              ;;SUPPRESS THIS 0?
12482  053366  001403                          BEQ     5$              ;;BR IF YES
12483  053370  005204                  4$:     INC     R4              ;;DON'T SUPPRESS ANYMORE 0'S
12484  053372  052703  000060                  BIS     #'0,R3          ;;MAKE THIS DIGIT ASCII
12485  053376  052703  000040          5$:     BIS     #' ,R3          ;;MAKE ASCII IF NOT ALREADY
12486  053402  110367  000040                  MOVB    R3,8$           ;;SAVE FOR TYPING
12487  053406  104400  053446                  TYPE    ,8$             ;;GO TYPE THIS DIGIT
12488  053412  105367  000032          7$:     DECB    $OCNT           ;;COUNT BY 1
12489  053416  003347                          BGT     2$              ;;BR IF MORE TO DO
12490  053420  002402                          BLT     6$              ;;BR IF DONE
12491  053422  005204                          INC     R4              ;;INSURE LAST DIGIT ISN'T A BLANK
```

```
12492  053424  000744                          BR      2$              ;;GO DO THE LAST DIGIT
12493  053426  012605                  6$:     MOV     (SP)+,R5        ;;RESTORE R5
12494  053430  012604                          MOV     (SP)+,R4        ;;RESTORE R4
12495  053432  012603                          MOV     (SP)+,R3        ;;RESTORE R3
12496  053434  016666  000002  000004          MOV     2(SP),4(SP)     ;;SET THE STACK FOR RETURNING
12497  053442  012616                          MOV     (SP)+,(SP)
12498  053444  000002                          RTI                     ;;RETURN
12499  053446     000                  8$:     .BYTE   0               ;;STORAGE FOR ASCII DIGIT
12500  053447     000                          .BYTE   0               ;;TERMINATOR FOR TYPE ROUTINE
12501  053450     000                  $OCNT:  .BYTE   0               ;;OCTAL DIGIT COUNTER
12502  053451     000                  $OFILL: .BYTE   0               ;;ZERO FILL SWITCH
12503  053452  000000                  $OMODE: .WORD   0               ;;NUMBER OF DIGITS TO TYPE
```

```
12504                                   .SBTTL  TRAP DECODER
12505
12506
12507                                   ;;*****************************************************************
12508                                   ;*THIS ROUTINE WILL PICKUP THE LOWER BYTE OF THE "TRAP" INSTRUCTION
12509                                   ;*AND USE IT TO INDEX THROUGH THE TRAP TABLE FOR THE STARTING ADDRESS
12510                                   ;*OF THE DESIRED ROUTINE. THEN USING THE ADDRESS OBTAINED IT WILL
12511                                   ;*GO TO THAT ROUTINE.
12512  053454  010046                   $TRAP:  MOV     R0,-(SP)        ;;SAVE R0
12513  053456  016600  000002                   MOV     2(SP),R0        ;;GET TRAP ADDRESS
12514  053462  005740                           TST     -(R0)           ;;BACKUP BY 2
12515  053464  111000                           MOVB    (R0),R0         ;;GET RIGHT BYTE OF TRAP
12516  053466  006300                           ASL     R0              ;;POSITION FOR INDEXING
12517  053470  016000  053476                   MOV     $TRPAD(R0),R0   ;;INDEX TO TABLE
12518  053474  000200                           RTS     R0              ;;GO TO ROUTINE
12519
12520                                   .SBTTL  TRAP TABLE
12521
12522                                   ;*THIS TABLE CONTAINS THE STARTING ADDRESSES OF THE ROUTINES CALLED
12523                                   ;*BY THE "TRAP" INSTRUCTION.
12524
12525                                   ;       ROUTINE
12526                                   ;       -------
12527  053476                           $TRPAD:
12528  053476  051570                           $TYPE   ;;CALL=TYPE     TRAP+0(104400)  TTY TYPEOUT ROUTINE
12529  053500  053252                           $TYPOC  ;;CALL=TYPOC    TRAP+1(104401)  TYPE OCTAL NUMBER (WITH LEADING ZEROS)
12530  053502  053226                           $TYPOS  ;;CALL=TYPOS    TRAP+2(104402)  TYPE OCTAL NUMBER (NO LEADING ZEROS)
12531  053504  053266                           $TYPON  ;;CALL=TYPON    TRAP+3(104403)  TYPE OCTAL NUMBER (AS PER LAST CALL)
12532  053506  051344                           $TYPDS  ;;CALL=TYPDS    TRAP+4(104404)  TYPE DECIMAL NUMBER (WITH SIGN)
12533
12534
12535  053510  052300                           $RDCHR  ;;CALL=RDCHR    TRAP+5(104405)  TTY TYPEIN CHARACTER ROUTINE
12536  053512  052370                           $RDLIN  ;;CALL=RDLIN    TRAP+6(104406)  TTY TYPEIN STRING ROUTINE
12537  053514  052550                           $RDOCT  ;;CALL=RDOCT    TRAP+7(104407)  READ AN OCTAL NUMBER FROM TTY
12538  053516  040046                           T.SCOP  ;;CALL=SCOP1    TRAP+10(104410) MY LOCAL SCOPES
12539  053520  040120                           CHECKT  ;;CALL=CHECKD   TRAP+11(104411) CHECK DVA,RDY,DPR,DRY
12540  053522  040422                           WAIT.T  ;;CALL=WAT      TRAP+12(104412) WAIT LOOP
12541
12542
12543
```

```
12544                                           .SBTTL  POWER DOWN AND UP ROUTINES
12545
12546                                           ;;*****************************************************************
12547                                           ;POWER DOWN ROUTINE
12548  053524  012737  053670  000024  $PWRDN:  MOV     #$ILLUP,@#PWRVEC ;;SET FOR FAST UP
12549  053532  012737  000340  000026          MOV     #340,@#PWRVEC+2  ;;PRIO:7
12550  053540  010046                          MOV     R0,-(SP)         ;;PUSH R0 ON STACK
12551  053542  010146                          MOV     R1,-(SP)         ;;PUSH R1 ON STACK
12552  053544  010246                          MOV     R2,-(SP)         ;;PUSH R2 ON STACK
12553  053546  010346                          MOV     R3,-(SP)         ;;PUSH R3 ON STACK
12554  053550  010446                          MOV     R4,-(SP)         ;;PUSH R4 ON STACK
12555  053552  010546                          MOV     R5,-(SP)         ;;PUSH R5 ON STACK
12556  053554  017746  125360                  MOV     @SWR,-(SP)       ;;PUSH @SWR ON STACK
12557  053560  010667  000110                  MOV     SP,$SAVR6        ;;SAVE SP
12558  053564  012737  053576  000024          MOV     #$PWRUP,@#PWRVEC ;;SET UP VECTOR
12559  053572  000000                          HALT
12560  053574  000776                          BR      .-2              ;;HANG UP
12561
12562                                           ;;*****************************************************************
12563                                           ;POWER UP ROUTINE
12564  053576  012737  053670  000024  $PWRUP:  MOV     #$ILLUP,@#PWRVEC ;;SET FOR FAST DOWN
12565  053604  016706  000064                  MOV     $SAVR6,SP        ;;GET SP
12566  053610  005067  000060                  CLR     $SAVR6           ;;WAIT LOOP FOR THE TTY
12567  053614  005267  000054          1$:     INC     $SAVR6           ;;WAIT FOR THE INC
12568  053620  001375                          BNE     1$               ;;OF  WORD
12569  053622  012677  125312                  MOV     (SP)+,@SWR       ;;POP STACK INTO @SWR
12570  053626  012605                          MOV     (SP)+,R5         ;;POP STACK INTO R5
12571  053630  012604                          MOV     (SP)+,R4         ;;POP STACK INTO R4
12572  053632  012603                          MOV     (SP)+,R3         ;;POP STACK INTO R3
12573  053634  012602                          MOV     (SP)+,R2         ;;POP STACK INTO R2
12574  053636  012601                          MOV     (SP)+,R1         ;;POP STACK INTO R1
12575  053640  012600                          MOV     (SP)+,R0         ;;POP STACK INTO R0
12576  053642  012737  053524  000024          MOV     #$PWRDN,@#PWRVEC ;;SET UP THE POWER DOWN VECTOR
12577  053650  012737  000340  000026          MOV     #340,@#PWRVEC+2  ;;PRIO:7
12578  053656  104400                          TYPE                     ;REPORT THE POWER FAILURE
12579  053660  053676          $PWRMG:  .WORD   $POWER                   ;;POWER FAIL MESSAGE POINTER
12580  053662  012716                          MOV     (PC)+,(SP)       ;;RESTART AT BEGIN !
12581  053664  004216          $PWRAD:  .WORD   BEGIN                    ;;RESTART ADDRESS
12582  053666  000002                          RTI
12583  053670  000000          $ILLUP:  HALT                             ;;THE POWER UP SEQUENCE WAS STARTED
12584  053672  000776                          BR      .-2              ;; BEFORE THE POWER DOWN WAS COMPLETE
12585  053674  000000          $SAVR6:  0                                ;;PUT THE SP HERE
12586  053676  005015  047520  042527  $POWER:  .ASCIZ  <15><12>"POWER"
12587  053704  000122
12588                                           .EVEN
```

```
12589                                   ;*****************************************************************************
12590                                   ;
12591                                   ;ERROR AND MESSAGE TABLE CONDIMENTS
12592                                   ;
12593                                   ;*****************************************************************************
12594
12595
12596
12597
12598  053706  051127  047117  020107  EM1:    .ASCIZ  /WRONG DATA IN READING OR WRITING HARDWARE REGISTER/
12599  053714  040504  040524  044440
12600  053722  020116  042522  042101
12601  053730  047111  020107  051117
12602  053736  053440  044522  044524
12603  053744  043516  044040  051101
12604  053752  053504  051101  020105
12605  053760  042522  044507  052123
12606  053766  051105     000
12607  053771     105  051122  051117  EM2:    .ASCIZ  /ERROR ON  DATA COMMAND/
12608  053776  047440  020116  042040
12609  054004  052101  020101  047503
12610  054012  046515  047101  000104
12611  054020  051105  047522  020122  EM6:    .ASCIZ  /ERROR ON WRITE HEADER AND DATA/
12612  054026  047117  053440  044522
12613  054034  042524  044040  040505
12614  054042  042504  020122  047101
12615  054050  020104  040504  040524
12616  054056     000
12617  054057     103  047117  051124  EM11:   .ASCIZ  /CONTROLLER OR DRIVE STATUS/
12618  054064  046117  042514  020122
12619  054072  051117  042040  044522
12620  054100  042526  051440  040524
12621  054106  052524  000123
12622  054112  042522  044507  052123  EM14:   .ASCIZ  /REGISTER FAILED/
12623  054120  051105  043040  044501
12624  054126  042514  000104
12625  054132  050123  041505  043111  EM15:   .ASCIZ  /SPECIFIED REGISTER NON EXISTANT SO ABORT/
12626  054140  042511  020104  042522
12627  054146  044507  052123  051105
12628  054154  047040  047117  042440
12629  054162  044530  052123  047101
12630  054170  020124  047523  040440
12631  054176  047502  052122     000
12632  054203     127  044501  020124  EM16:   .ASCIZ  /WAIT LOOP FAILED/
12633  054210  047514  050117  043040
12634  054216  044501  042514  000104
12635  054224  051127  052111  020105  EM17:   .ASCIZ  /WRITE CHECK FAILING/
12636  054232  044103  041505  020113
12637  054240  040506  046111  047111
12638  054246  000107
12639  054250  042522  044507  052123  EM20:   .ASCIZ  /REGISTER FAILING/
12640  054256  051105  043040  044501
12641  054264  044514  043516     000
12642  054271     111  052116  051105  EM21:   .ASCIZ  /INTERRUPT FAILING/
12643  054276  052522  052120  040040
12644  054304  044501  044514  043516
```

```
12645  054312      000
12646  054313      105  051122  051117  EM22:   .ASCII  /ERROR ON DRIVE PRESENT/<15><12>
12647  054320  047440  020116  051104
12648  054326  053111  020105  051120
12649  054334  051505  047105  006524
12650  054342      012
12651  054343      124  042510  052440          .ASCII  /THE UNIT NO. FOUND BY SETTING RHAS/<15><12>
12652  054350  044516  020124  047516
12653  054356  020056  047506  047125
12654  054364  020104  054502  051440
12655  054372  052105  044524  043516
12656  054400  051040  040510  006523
12657  054406      012
12658  054407      104  020117  047516          .ASCII  /DO NOT AGREE WITH THE UNIT NO. FOUND FROM/<15><12>
12659  054414  020124  043501  042522
12660  054422  020105  044527  044124
12661  054430  052040  042510  052440
12662  054436  044516  020124  047516
12663  054444  020056  047506  047125
12664  054452  020104  051106  046517
12665  054460  005015
12666  054462  044122  051503  026462          .ASCII  /RHCS2-NED BIT #12/<15><12>
12667  054470  042516  020104  044502
12668  054476  020124  030443  006462
12669  054504      012
12670  054505      061  033467  033467          .ASCII  /177777-MEANS NO UNIT FOUND/<15><12>
12671  054512  026467  042515  047101
12672  054520  020123  047516  052440
12673  054526  044516  020124  047506
12674  054534  047125  006504      012
12675  054541      116  052117  035105          .ASCII  /NOTE: ON DUAL PORT SYSTEM, DRIVE ON OTHER PORT WILL NOT GIVE/<15><12>
12676  054546  047440  020116  052504
12677  054554  046101  050040  051117
12678  054562  020124  054523  052123
12679  054570  046505  020054  051104
12680  054576  053111  020105  047117
12681  054604  047440  044124  051105
12682  054612  050040  051117  020124
12683  054620  044527  046114  047040
12684  054626  052117  043440  053111
12685  054634  006505      012
12686  054637      116  042105  044040          .ASCIZ  /NED HENCE THERE WILL BE AN EXTRA DRIVE/
12687  054644  047105  042503  052040
12688  054652  042510  042522  053440
12689  054660  046111  020114  042502
12690  054666  040440  020116  054105
12691  054674  051124  020101  051104
12692  054702  053111  000105
12693  054706  047514  045517  040440  EM24:   .ASCIZ  /LOOK AHEAD REGISTER AT THE BEGINNING OF SECTOR IS IN ERROR/
12694  054714  042510  042101  051040
12695  054722  043505  051511  042524
12696  054730  020122  052101  052040
12697  054736  042510  041040  043505
12698  054744  047111  044516  043516
12699  054752  047440  020106  042523
12700  054760  052103  051117  044440
```

```
12701  054766  020123  047111  042440
12702  054774  051122  051117     000
12703  055001     114  047517  020113  EM25:   .ASCIZ  /LOOK AHEAD REGISTER IS IN ERROR/
12704  055006  044101  040505  020104
12705  055014  042522  044507  052123
12706  055022  051105  044440  020123
12707  055030  047111  042440  051122
12708  055036  051117     000
12709  055041     103  051125  042522  EM30:   .ASCII  /CURRENT CYLINDER DOES NOT MATCH DESIRED CYLINDER REGIHSTER/<15><12>
12710  055046  052116  041440  046131
12711  055054  047111  042504  020122
12712  055062  047504  051505  047040
12713  055070  052117  046440  052101
12714  055076  044103  042040  051505
12715  055104  051111  042105  041440
12716  055112  046131  047111  042504
12717  055120  020122  042522  044507
12718  055126  051510  042524  006522
12719  055134     012
12720  055135     101  052106  051105          .ASCIZ  /AFTER A SEEK AND INIT/
12721  055142  040440  051440  042505
12722  055150  020113  047101  020104
12723  055156  047111  052111     000
12724  055163     105  041503  043440  EM31:   .ASCII  /ECC GENERATED IS INCORRECT/<15><12>
12725  055170  047105  051105  052101
12726  055176  042105  044440  020123
12727  055204  047111  047503  051122
12728  055212  041505  006524     012
12729  055217     105  042526  054522          .ASCIZ  /EVERY WORD ON THIS SECTOR IS THAT GIVEN IN "DATA USED"/
12730  055224  053440  051117  020104
12731  055232  047117  052040  044510
12732  055240  020123  042523  052103
12733  055246  051117  044440  020123
12734  055254  044124  052101  043440
12735  055262  053111  047105  044440
12736  055270  020116  042042  052101
12737  055276  020101  051525  042105
12738  055304  000042
12739  055306  047117  051040  040505  EM32:   .ASCII  /ON READ COMMAND AFTER DATA AND ECC HAVE BEEN READ/<15><12>
12740  055314  020104  047503  046515
12741  055322  047101  020104  043101
12742  055330  042524  020122  040504
12743  055336  040524  040440  042116
12744  055344  042440  041503  044040
12745  055352  053101  020105  042502
12746  055360  047105  051040  040505
12747  055366  006504     012
12748  055371     105  041503  051040          .ASCII  /ECC REGISTERS OR RHER1 IS IN ERROR/<15><12>
12749  055376  043505  051511  042524
12750  055404  051522  047440  020122
12751  055412  044122  051105  020061
12752  055420  051511  044440  020116
12753  055426  051105  047522  006522
12754  055434     012
12755  055435     117  046116  020131          .ASCII  /ONLY LOWER 11 BITS OF PATTERN REG. CAN BE READ/<15><12>
12756  055442  047514  042527  020122
```

```
 12757  055450  030461  041040  052111
 12758  055456  020123  043117  050040
 12759  055464  052101  042524  047122
 12760  055472  051040  043505  020056
 12761  055500  040503  020116  042502
 12762  055506  051040  040505  006504
 12763  055514     012
 12764  055515     124  044510  020123          .ASCIZ  /THIS SHOULD MATCH LOWER 11 BITS OF GOOD ECC1/
 12765  055522  044123  052517  042114
 12766  055530  046440  052101  044103
 12767  055536  046040  053517  051105
 12768  055544  030440  020061  044502
 12769  055552  051524  047440  020106
 12770  055560  047507  042117  042440
 12771  055566  041503  000061
 12772  055572  044510  044107  041440  EM33:   .ASCIZ  /HIGH COUNT BIT NOT SET AFTER 38859 CLOCKS/
 12773  055600  052517  052116  041040
 12774  055606  052111  047040  052117
 12775  055614  051440  052105  040440
 12776  055622  052106  051105  031440
 12777  055630  034070  034465  041440
 12778  055636  047514  045503  000123
 12779  055644  042532  047522  042040  EM34:   .ASCIZ  /ZERO DETECT BIT NOT HIGH WHEN 32 BIT ECC REG. HAS 21 ZEROS/
 12780  055652  052105  041505  020124
 12781  055660  044502  020124  047516
 12782  055666  020124  044510  044107
 12783  055674  053440  042510  020116
 12784  055702  031063  041040  052111
 12785  055710  042440  041503  051040
 12786  055716  043505  020056  040510
 12787  055724  020123  030462  055040
 12788  055732  051105  051517     000
 12789  055737     120  051517  052111  EM35:   .ASCII  /POSITION REGISTER OR 11 BITS OF PATTERN REGISTER INCORRECT/<15><12>
 12790  055744  047511  020116  042522
 12791  055752  044507  052123  051105
 12792  055760  047440  020122  030461
 12793  055766  041040  052111  020123
 12794  055774  043117  050040  052101
 12795  056002  042524  047122  051040
 12796  056010  043505  051511  042524
 12797  056016  020122  047111  047503
 12798  056024  051122  041505  006524
 12799  056032     012
 12800  056033     114  053517  051105          .ASCII  /LOWER 11 BITS OF PATTERN REGISTER SHOULD MATCH LOWER/<15><12>
 12801  056040  030440  020061  044502
 12802  056046  051524  047440  020106
 12803  056054  040520  052124  051105
 12804  056062  020116  042522  044507
 12805  056070  052123  051105  051440
 12806  056076  047510  046125  020104
 12807  056104  040515  041524  020110
 12808  056112  047514  042527  006522
 12809  056120     012
 12810  056121     061  020061  044502          .ASCII  /11 BITS OF GOOD ECC1/<15><12>
 12811  056126  051524  047440  020106
 12812  056134  047507  042117  042440
```

```
12813  056142  041503  006461     012
12814  056147     104  052101  042440          .ASCIZ  /DAT ENVLOP GOOD POSITION AND N-CODE ZEROS ARE IN OCTAL/
12815  056154  053116  047514  020120
12816  056162  047507  042117  050040
12817  056170  051517  052111  047511
12818  056176  020116  047101  020104
12819  056204  026516  047503  042504
12820  056212  055040  051105  051517
12821  056220  040440  042522  044440
12822  056226  020116  041517  040524
12823  056234  000114
12824  056236  047117  051040  040505  EM36:   .ASCII  /ON READ COMMAND WITH NON-CORRECTABLE ERROR DCK AND ECH SHOULD BE SET/<1
12825  056244  020104  047503  046515
12826  056252  047101  020104  044527
12827  056260  044124  047040  047117
12828  056266  041455  051117  042522
12829  056274  052103  041101  042514
12830  056302  042440  051122  051117
12831  056310  042040  045503  040440
12832  056316  042116  042440  044103
12833  056324  051440  047510  046125
12834  056332  020104  042502  051440
12835  056340  052105  005015
12836  056344  051120  043517  040522  EM37:   .ASCIZ  /PROGRAM ERROR BIT #10 IN RHCS2 DID NOT SET/
12837  056352  020115  051105  047522
12838  056360  020122  044502  020124
12839  056366  030443  020060  047111
12840  056374  051040  041510  031123
12841  056402  042040  042111  047040
12842  056410  052117  051440  052105
12843  056416     000
12844  056417     111  020106  047520          .ASCIZ  /IF POSITION REGISTER =10040 OR 10041 IT IS GOOD/
12845  056424  044523  044524  047117
12846  056432  051040  043505  051511
12847  056440  042524  020122  030475
12848  056446  030060  030064  047440
12849  056454  020122  030061  032060
12850  056462  020061  052111  044440
12851  056470  020123  047507  042117
12852  056476     000
12853  056477     040  050040  020103  DH1:    .ASCII  /  PC      TEST    REG.    GOOD  RECEIVED/<15><12>
12854  056504  020040  020040  052040
12855  056512  051505  020124  020040
12856  056520  051040  043505  020056
12857  056526  020040  043440  047517
12858  056534  020104  051040  041505
12859  056542  044505  042526  006504
12860  056550     012
12861  056551     040  020040  020040          .ASCIZ  /          NO    ADDR.    DATA    DATA  /
12862  056556  020040  020040  020040
12863  056564  047516  020040  020040
12864  056572  042101  051104  020056
12865  056600  020040  042040  052101
12866  056606  020101  020040  042040
12867  056614  052101  020101  000040
12868  056622  020040  041520  020040  DH2:    .ASCII  /  PC    TEST      WORD    GOOD     BAD  /<15><12>
```

```
12869  056630  020040  042524  052123
12870  056636  020040  020040  020040
12871  056644  047527  042122  020040
12872  056652  020040  047507  042117
12873  056660  020040  020040  041040
12874  056666  042101  020040  005015
12875  056674  020040  020040  020040          .ASCIZ  /        NO      NO     DATA    DATA  /
12876  056702  020040  047040  020117
12877  056710  020040  020040  020040
12878  056716  047040  020117  020040
12879  056724  020040  040504  040524
12880  056732  020040  020040  040504
12881  056740  040524  020040     000
12882  056745     040  050040  020103  DH11:   .ASCII  /  PC      TEST  FAILING  CONT.   CONT.   CONT.   CONT. /<15><12>
12883  056752  020040  020040  052040
12884  056760  051505  020124  043040
12885  056766  044501  044514  043516
12886  056774  020040  047503  052116
12887  057002  020056  020040  047503
12888  057010  052116  020056  020040
12889  057016  047503  052116  020056
12890  057024  020040  047503  052116
12891  057032  020056  005015
12892  057036  020040  020040  020040          .ASCIZ  /          NO   REG ADR RHCS1   RHCS2   RHDS1   RHER1/
12893  057044  020040  020040  047040
12894  057052  020117  020040  042522
12895  057060  020107  042101  020122
12896  057066  044122  051503  020061
12897  057074  020040  044122  051503
12898  057102  020062  020040  044122
12899  057110  051504  020061  020040
12900  057116  044122  051105  000061
12901  057124  020040  050040  020103  DH14:   .ASCII  /  PC      TEST  FAILING   CONT.   CONT.   CONT.   CONT.   CONT. /<15><1
12902  057132  020040  020040  042524
12903  057140  052123  020040  040506
12904  057146  046111  047111  020107
12905  057154  020040  047503  052116
12906  057162  020056  020040  047503
12907  057170  052116  020056  020040
12908  057176  047503  052116  020056
12909  057204  020040  047503  052116
12910  057212  020056  020040  047503
12911  057220  052116  020056  005015
12912  057226  020040  020040  020040          .ASCIZ  /          NO   REG ADR  BAD REG  RHCS1   RHCS2   RHDS1   RHER1/
12913  057234  020040  020040  047040
12914  057242  020117  020040  042522
12915  057250  020107  042101  020122
12916  057256  041040  042101  051040
12917  057264  043505  020040  044122
12918  057272  051503  020061  020040
12919  057300  044122  051503  020062
12920  057306  020040  044122  051504
12921  057314  020061  020040  044122
12922  057322  051105  000061
12923  057326  020040  050040  020103  DH15:   .ASCIZ  /  PC   REG. ADR./
12924  057334  020040  042522  027107
```

```
12925  057342  040440  051104  000056
12926  057350  020040  041520  020040  DH16:   .ASCII  /  PC     WAT    BIT    REG    REG/<15><12>
12927  057356  020040  020040  040527
12928  057364  020124  020040  041040
12929  057372  052111  020040  020040
12930  057400  051040  043505  020040
12931  057406  020040  051040  043505
12932  057414  005015
12933  057416  020040  020040  020040          .ASCIZ  /        PC    WANTED  ADDRESS CONTENTS/
12934  057424  020040  020040  041520
12935  057432  020040  020040  040527
12936  057440  052116  042105  020040
12937  057446  042101  051104  051505
12938  057454  020123  047503  052116
12939  057462  047105  051524     000
12940  057467     040  020040  041520  DH17:   .ASCII  /  PC     TEST    CONT.   CONT.   CONT.   CONT.   CONT. /<15><12>
12941  057474  020040  020040  052040
12942  057502  051505  020124  020040
12943  057510  041440  047117  027124
12944  057516  020040  041440  047117
12945  057524  027124  020040  041440
12946  057532  047117  027124  020040
12947  057540  041440  047117  027124
12948  057546  020040  041440  047117
12949  057554  027124  006440     012
12950  057561     040  020040  020040          .ASCIZ  /         NO     RHBA    RHDB    RHWC    RHCS1   RHCS2 /
12951  057566  020040  020040  020040
12952  057574  047516  020040  020040
12953  057602  051040  041110  020101
12954  057610  020040  051040  042110
12955  057616  020102  020040  051040
12956  057624  053510  020103  020040
12957  057632  051040  041510  030523
12958  057640  020040  051040  041510
12959  057646  031123  000040
12960  057652  020040  050040  020103  DH20:   .ASCII  /  PC   TEST    CONT    CONT    CONT    CONT    CONT/<15><12>
12961  057660  020040  042524  052123
12962  057666  020040  020040  047503
12963  057674  052116  020040  020040
12964  057702  047503  052116  020040
12965  057710  020040  047503  052116
12966  057716  020040  020040  047503
12967  057724  052116  020040  020040
12968  057732  047503  052116  005015
12969  057740  020040  020040  020040          .ASCIZ  /         NO     RHER1   RHER2   RHER3   RHAS    RHDS1/
12970  057746  020040  047040  020117
12971  057754  020040  020040  044122
12972  057762  051105  020061  020040
12973  057770  044122  051105  020062
12974  057776  020040  044122  051105
12975  060004  020063  020040  044122
12976  060012  051501  020040  020040
12977  060020  044122  051504  000061
12978  060026  020040  050040  020103  DH21:   .ASCII  /  PC   TEST    CONT    CONT    CONT  /<15><12>
12979  060034  020040  052040  051505
12980  060042  020124  020040  041440
```

```
 12981  060050  047117  020124  020040
 12982  060056  041440  047117  020124
 12983  060064  020040  041440  047117
 12984  060072  020124  006440     012
 12985  060077     040  020040  020040          .ASCIZ  /          NO     RHCS1   RHAS   RHDS1 /
 12986  060104  020040  020040  020040
 12987  060112  047516  020040  020040
 12988  060120  051040  041510  030523
 12989  060126  020040  051040  040510
 12990  060134  020123  020040  044122
 12991  060142  051504  020061     000
 12992  060147     040  050040  020103  DH22:   .ASCII  /  PC    TEST    RHAS    RHCS2   /<15><12>
 12993  060154  020040  052040  051505
 12994  060162  020124  020040  051040
 12995  060170  040510  020123  020040
 12996  060176  051040  041510  031123
 12997  060204  020040  006440     012
 12998  060211     040  020040  020040          .ASCIZ  /        NO      UNIT    UNIT/
 12999  060216  020040  047040  020117
 13000  060224  020040  020040  052440
 13001  060232  044516  020124  020040
 13002  060240  052440  044516  000124
 13003  060246  041520  020040  020040  DH24:   .ASCII  /PC      RHDST   BAD     GOOD    SECTOR  SECTOR/<15><12>
 13004  060254  020040  044122  051504
 13005  060262  020124  020040  040502
 13006  060270  020104  020040  020040
 13007  060276  047507  042117  020040
 13008  060304  020040  042523  052103
 13009  060312  051117  020040  042523
 13010  060320  052103  051117  005015
 13011  060326  020040  020040  020040          .ASCIZ  /        CONT.   RHLA    RHLA    NO      CLOCK/
 13012  060334  020040  047503  052116
 13013  060342  020056  020040  044122
 13014  060350  040514  020040  020040
 13015  060356  044122  040514  020040
 13016  060364  020040  047516  020040
 13017  060372  020040  020040  046103
 13018  060400  041517  000113
 13019  060404  041520  020040  020040  DH26:   .ASCII  /PC      PC OF   FAILING CONT.    CONT.    CONT.    CONT./<15><12>
 13020  060412  020040  041520  047440
 13021  060420  020106  020040  040506
 13022  060426  046111  047111  020107
 13023  060434  047503  052116  020056
 13024  060442  020040  041440  047117
 13025  060450  027124  020040  020040
 13026  060456  047503  052116  020056
 13027  060464  020040  041440  047117
 13028  060472  027124  005015
 13029  060476  020040  020040  020040          .ASCIZ/        JSR     REG. ADDR RHCS1   RHCS2   RHDS1   RHER1/
 13030  060504  020040  051512  020122
 13031  060512  020040  020040  042522
 13032  060520  027107  040440  042104
 13033  060526  020122  044122  051503
 13034  060534  020061  020040  044122
 13035  060542  051503  020062  020040
 13036  060550  044122  051504  020061
```

```
13037  060556  020040  044122  051105
13038  060564  000061
13039  060566  041520  020040  020040  DH27:   .ASCII  /PC     PC OF   TEST    FAILING GOOD    RECEIVED/<15><12>
13040  060574  020040  041520  047440
13041  060602  020106  020040  042524
13042  060610  052123  020040  020040
13043  060616  040506  046111  047111
13044  060624  020107  047507  042117
13045  060632  020040  020040  042522
13046  060640  042503  053111  042105
13047  060646  005015
13048  060650  020040  020040  020040          .ASCIZ  /       JSR     NO      REGISTERDATA    DATA/
13049  060656  020040  051512  020122
13050  060664  020040  020040  047040
13051  060672  020117  020040  020040
13052  060700  042522  044507  052123
13053  060706  051105  040504  040524
13054  060714  020040  020040  040504
13055  060722  040524     000
13056  060725     120  020103  020040  DH30:   .ASCII  /PC     PC OF   REGISTER GOOD   BAD/<15><12>
13057  060732  020040  050040  020103
13058  060740  043117  020040  051040
13059  060746  043505  051511  042524
13060  060754  020122  047507  042117
13061  060762  020040  041040  042101
13062  060770  005015
13063  060772  020040  020040  020040          .ASCIZ  /       JSR     ADDRESS DATA    DATA/
13064  061000  020040  051512  020122
13065  061006  020040  020040  042101
13066  061014  051104  051505  020123
13067  061022  040504  040524  020040
13068  061030  042040  052101  000101
13069  061036  020040  041520  020040  DH31:   .ASCII  / PC    TEST     WORD    GOOD     BAD  CONT.   CONT.   CONT   /<15><12
13070  061044  020040  042524  052123
13071  061052  020040  020040  020040
13072  061060  047527  042122  020040
13073  061066  020040  047507  042117
13074  061074  020040  020040  041040
13075  061102  042101  020040  047503
13076  061110  052116  020056  020040
13077  061116  047503  052116  020056
13078  061124  020040  047503  052116
13079  061132  020040  006440     012
13080  061137     040  020040  020040          .ASCIZ  /       NO      NO      DATA    DATA  RHCS1   RHDS1   RHER1   /
13081  061144  020040  020040  047516
13082  061152  020040  020040  020040
13083  061160  020040  047516  020040
13084  061166  020040  042040  052101
13085  061174  020101  020040  042040
13086  061202  052101  020101  051040
13087  061210  041510  030523  020040
13088  061216  051040  042110  030523
13089  061224  020040  051040  042510
13090  061232  030522  020040  000040
13091  061240  020040  041520  020040  DH32:   .ASCII  / PC    PC OF   TEST     WORD    GOOD     BAD  CONT.   CONT.   CONT.
13092  061246  020040  041520  047440
```

```
13093  061254  020106  020040  042524
13094  061262  052123  020040  020040
13095  061270  020040  047527  042122
13096  061276  020040  020040  047507
13097  061304  042117  020040  020040
13098  061312  041040  042101  020040
13099  061320  047503  052116  020056
13100  061326  020040  047503  052116
13101  061334  020056  020040  047503
13102  061342  052116  020056  020040
13103  061350  005015
13104  061352  020040  020040  020040          .ASCIZ  /      JSR     NO      NO      DATA    DATA    RHCS1   RHDS1   RHER1
13105  061360  020040  045040  051123
13106  061366  020040  020040  047040
13107  061374  020117  020040  020040
13108  061402  020040  047040  020117
13109  061410  020040  020040  040504
13110  061416  040524  020040  020040
13111  061424  040504  040524  020040
13112  061432  044122  051503  020061
13113  061440  020040  044122  051504
13114  061446  020061  020040  044122
13115  061454  051105  020061  020040
13116  061462     000
13117  061463     040  050040  020103  DH33:   .ASCII  /  PC    PC OF   TEST    WORD    GOOD    CONT.   CONT.   CONT.   /<15
13118  061470  020040  050040  020103
13119  061476  043117  020040  052040
13120  061504  051505  020124  020040
13121  061512  020040  053440  051117
13122  061520  020104  020040  043440
13123  061526  047517  020104  020040
13124  061534  020040  047503  052116
13125  061542  020056  020040  047503
13126  061550  052116  020056  020040
13127  061556  047503  052116  020056
13128  061564  020040  005015
13129  061570  020040  020040  020040          .ASCIZ  /      JSR     NO      NO      DATA    RHCS1   RHDS1   RHER1   /
13130  061576  020040  045040  051123
13131  061604  020040  020040  047040
13132  061612  020117  020040  020040
13133  061620  020040  047040  020117
13134  061626  020040  020040  040504
13135  061634  040524  020040  020040
13136  061642  051040  041510  030523
13137  061650  020040  051040  042110
13138  061656  030523  020040  051040
13139  061664  042510  030522  020040
13140  061672  000040
13141  061674  041520  020040  020040  DH34:   .ASCII  /PC     TEST    GOOD    GOOD    WRITTEN WRITTEN DATA/<15><12>
13142  061702  020040  042524  052123
13143  061710  020040  020040  047507
13144  061716  042117  020040  020040
13145  061724  047507  042117  020040
13146  061732  020040  051127  052111
13147  061740  042524  020116  051127
13148  061746  052111  042524  020116
```

```
13149  061754  040504  040524  005015
13150  061762  020040  020040  020040          .ASCIZ  /       NO    ECC1   ECC2   ECC1   ECC2   USED/
13151  061770  020040  047040  020117
13152  061776  020040  020040  041505
13153  062004  030503  020040  020040
13154  062012  041505  031103  020040
13155  062020  020040  041505  030503
13156  062026  020040  020040  041505
13157  062034  031103  020040  020040
13158  062042  051525  042105     000
13159  062047     120  020103  020040  DH35:  .ASCII  /PC     TEST   GOOD   GOOD   PATTERN POSITON GOOD   RHER1/<15><12>
13160  062054  020040  052040  051505
13161  062062  020124  020040  043440
13162  062070  047517  020104  020040
13163  062076  043440  047517  020104
13164  062104  020040  050040  052101
13165  062112  042524  047122  050040
13166  062120  051517  052111  047117
13167  062126  043440  047517  020104
13168  062134  020040  051040  042510
13169  062142  030522  005015
13170  062146  020040  020040  020040          .ASCIZ  /       NO    ECC1   ECC2   REG.   REG.   POSITON REG./
13171  062154  020040  047040  020117
13172  062162  020040  020040  041505
13173  062170  030503  020040  020040
13174  062176  041505  031103  020040
13175  062204  020040  042522  027107
13176  062212  020040  020040  042522
13177  062220  027107  020040  020040
13178  062226  047520  044523  047524
13179  062234  020116  042522  027107
13180  062242     000
13181  062243     120  020103  020040  DH36:  .ASCII  /PC     PC OF  TEST   RHMR   POSITON PATTERN/<15><12>
13182  062250  020040  050040  020103
13183  062256  043117  020040  052040
13184  062264  051505  020124  020040
13185  062272  051040  046510  020122
13186  062300  020040  050040  051517
13187  062306  052111  047117  050040
13188  062314  052101  042524  047122
13189  062322  005015
13190  062324  020040  020040  020040          .ASCIZ  /       JSR    NO     CONT.  REG.   REG./
13191  062332  020040  045040  051123
13192  062340  020040  020040  047040
13193  062346  020117  020040  020040
13194  062354  047503  052116  020056
13195  062362  020040  042522  027107
13196  062370  020040  020040  042522
13197  062376  027107     000
13198  062401     120  020103  020040  DH37:  .ASCII  /PC     TEST   POSITON POSITON GOOD   GOOD   PATTERN DATA   N-CODE/
13199  062406  020040  052040  051505
13200  062414  020124  020040  050040
13201  062422  051517  052111  047117
13202  062430  050040  051517  052111
13203  062436  047117  043440  047517
13204  062444  020104  020040  043440
```

```
13205  062452  047517  020104  020040
13206  062460  050040  052101  042524
13207  062466  047122  042040  052101
13208  062474  020101  020040  047040
13209  062502  041455  042117  006505
13210  062510    012
13211  062511    040  020040  020040          .ASCIZ /       NO      ECC     GOOD    ECC1    ECC2     ECC   ENVLOPE ZEROS/
13212  062516  020040  020040  047516
13213  062524  020040  020040  020040
13214  062532  041505  020103  020040
13215  062540  020040  047507  042117
13216  062546  020040  020040  041505
13217  062554  030503  020040  020040
13218  062562  041505  031103  020040
13219  062570  020040  042440  041503
13220  062576  020040  042440  053116
13221  063604  047514  042520  055040
13222  062612  051105  051517     000
13223          062620                 .EVEN
13224  062620  001116  004172  037534  DT1:    .WORD   $ERRPC,TSTNM,REGADR,$GDDAT,$BDDAT,0
13225  062626  001124  001126  000000
13226  062634  001116  004172  044532  DT2:    .WORD   $ERRPC,TSTNM,ERWORD,$GDDAT,0
13227  062642  001124  000000
13228  062646  001116  004172  044532  DT3:    .WORD   $ERRPC,TSTNM,ERWORD,$GDDAT,$BDDAT,0
13229  062654  001124  001126  000000
13230  062662  001116  004172  001122  DT11:   .WORD   $ERRPC,TSTNM,$BDADR,CS1,CS2,DS1,ER1,0
13231  062670  001700  001676  001722
13232  062676  001702  000000
13233  062702  001116  004172  001122  DT14:   .WORD   $ERRPC,TSTNM,$BDADR,$BDDAT,CS1,CS2,DS1,ER1,0
13234  062710  001126  001700  001676
13235  062716  001722  001702  000000
13236  062724  001116  001200  000000  DT15:   .WORD   $ERRPC,$TMP1,0
13237  062732  001116  001204  001200  DT16:   .WORD   $ERRPC,$TMP3,$TMP1,$TMP0,$BDDAT,0
13238  062740  001176  001126  000000
13239  062746  001116  004172  001674  DT17:   .WORD   $ERRPC,TSTNM,BA,DB,WC,CS1,CS2,0
13240  062754  001670  001672  001700
13241  062762  001676  000000
13242  062766  001116  004172  001702  DT20:   .WORD   $ERRPC,TSTNM,ER1,ER2,ER3,AS,DS1,0
13243  062774  001706  001714  001716
13244  063002  001722  000000
13245  063006  001116  004172  001700  DT21:   .WORD   $ERRPC,TSTNM,CS1,AS,DS1,0
13246  063014  001716  001722  000000
13247  063022  001116  004172  001124  DT22:   .WORD   $ERRPC,TSTNM,$GDDAT,$BDDAT,0
13248  063030  001126  000000
13249  063034  001116  001704  001126  DT24:   .WORD   $ERRPC,DST,$BDDAT,$TMP1,$TMP2,$TMP3,0
13250  063042  001200  001202  001204
13251  063050  000000
13252  063052  001116  002000  001122  DT26:   .WORD   $ERRPC,PCJSR,$BDADR,CS1,CS2,DS1,ER1,0
13253  063060  001700  001676  001722
13254  063066  001702  000000
13255  063072  001116  002000  004172  DT27:   .WORD   $ERRPC,PCJSR,TSTNM,REGADR,$GDDAT,$BDDAT,0
13256  063100  037534  001124  001126
13257  063106  000000
13258  063110  001116  002000  037534  DT30:   .WORD   $ERRPC,PCJSR,REGADR,$GDDAT,$BDDAT,0
13259  063116  001124  001126  000000
13260
```

```
13261  063124  001116  004172  044532  DT31:   .WORD   $ERRPC,TSTNM,ERWORD,$GDDAT,$BDDAT,CS1,DS1,ER1,0
13262  063132  001124  001126  001700
13263  063140  001722  001702  000000
13264  063146  001116  002000  004172  DT32:   .WORD   $ERRPC,PCJSR,TSTNM,ERWORD,$GDDAT,$BDDAT,CS1,DS1,ER1,0
13265  063154  044532  001124  001126
13266  063162  001700  001722  001702
13267  063170  000000
13268  063172  001116  002000  004172  DT33:   .WORD   $ERRPC,PCJSR,TSTNM,ERWORD,$GDDAT,CS1,DS1,ER1,0
13269  063200  044532  001124  001700
13270  063206  001722  001702  000000
13271  063214  001116  004172  042522  DT34:   .WORD   $ERRPC,TSTNM,GECC1,GECC2,WECC1,WECC2,DISK,0
13272  063222  042524  047330  047332
13273  063230  046330  000000
13274  063234  001116  004172  042522  DT35:   .WORD   $ERRPC,TSTNM,GECC1,GECC2,EC2,EC1,POSITI,ER1,0
13275  063242  042524  001732  001730
13276  063250  042534  001702  000000
13277  063256  001116  002000  004172  DT36:   .WORD   $ERRPC,PCJSR,TSTNM,MR,EC1,EC2,0
13278  063264  001720  001730  001732
13279  063272  000000
13280  063274  001116  004172  001730  DT37:   .WORD   $ERRPC,TSTNM,EC1,POSITI,GECC1,GECC2,EC2,DATENV,ZCODE,0
13281  063302  042534  042522  042524
13282  063310  001732  042540  042542
13283  063316  000000
13284  063320     000     000     000  DF1:    .BYTE   0,0,0,0,0
13285  063323     000     000
13286  063325     000     000     001  DF2:    .BYTE   0,0,1,0
13287  063330     000
13288  063331     000     000     001  DF3:    .BYTE   0,0,1,0,0
13289  063334     000     000
13290  063336     000     000     000  DF11:   .BYTE   0,0,0,0,0,0,0
13291  063341     000     000     000
13292  063344     000
13293  063345     000     000     000  DF14:   .BYTE   0,0,0,0,0,0,0,0
13294  063350     000     000     000
13295  063353     000     000
13296  063355     000     000          DF15:   .BYTE   0,0
13297  063357     000     000     000  DF16:   .BYTE   0,0,0,0
13298  063362     000
13299  063363     000     000     000  DF17:   .BYTE   0,0,0,0,0,0,0
13300  063366     000     000     000
13301  063371     000
13302  063372     000     000     000  DF20:   .BYTE   0,0,0,0,0,0,0
13303  063375     000     000     000
13304  063400     000
13305  063401     000     000     000  DF21:   .BYTE   0,0,0,0,0
13306  063404     000     000
13307  063406     000     000     000  DF22:   .BYTE   0,0,0,0
13308  063411     000
13309  063412     000     000     000  DF24:   .BYTE   0,0,0,0,0,0
13310  063415     000     000     000
13311  063420     000     000     000  DF26:   .BYTE   0,0,0,0,0,0,0
13312  063423     000     000     000
13313  063426     000
13314  063427     000     000     000  DF27:   .BYTE   0,0,0,0,0,0
13315  063432     000     000     000
13316  063435     000     000     000  DF30:   .BYTE   0,0,0,0,0
```

```
13317  063440  000  000
13318  063442  000  000  001  DF31:  .BYTE  0,0,1,0,0,0,0,0
13319  063445  000  000  000
13320  063450  000  000
13321  063452  000  000  000  DF32:  .BYTE  0,0,0,1,0,0,0,0,0
13322  063455  001  000  000
13323  063460  000  000  000
13324  063463  000  000  000  DF33:  .BYTE  0,0,0,1,0,0,0,0
13325  063466  001  000  000
13326  063471  000  000
13327  063473  000  000  000  DF34:  .BYTE  0,0,0,0,0,0,0
13328  063476  000  000  000
13329  063501  000
13330  063502  000  000  000  DF35:  .BYTE  0,0,0,0,0,0,0,0
13331  063505  000  000  000
13332  063510  000  000
13333  063512  000  000  000  DF36:  .BYTE  0,0,0,0,0,0
13334  063515  000  000  000
13335  063520  000  000  000  DF37:  .BYTE  0,0,0,0,0,0,0,0,0
13336  063523  000  000  000
13337  063526  000  000  000
13338          063532                      .EVEN
13339
13340          000001                .END
```

DERPSB.P11 CROSS REFERENCE TABLE -- USER SYMBOLS

```
ACL    = 000040          2648#
ACU    = 100000          2620#
AOE    = 001000          2550#  9200
AS       001716          2729#  4733   4801   7532   7581  13242  13245
ATA    = 100000          2538#  6166   7541   7560   8268   9206   9358   9470   9578
ATABLE   004176          2822#  3137
ATTENT   002002          2769#  3135*  3137*  5013   6163   7521   7579   9204   9356   9468   9576
ATO    = 000001          2572#
AT1    = 000002          2573#
AT2    = 000004          2574#
AT3    = 000010          2575#
AT4    = 000020          2576#
AT5    = 000040          2577#
AT6    = 000100          2578#
AT7    = 000200          2579#
A16    = 000400          2513#  4629   7767   7784
A17    = 001000          2514#  4629   7767   7784
BA       001674          2718#  7914  13239
BAI    = 000010          2484#  4317   4322   7663   7776   7792   8348  10488
BASECH   043452          2961  10882#
BEGIN    004216          1903   2829#  12581
BEGIN2   004206          1905   2827#
BITST    037536          2988   3221   3247   3273   3300   3326   3414   3440   3466   3492   3518   9804#
BIT0   = 000001          1877#  7527  10233  10268  10270
BIT00  = 000001          1867#  1877
BIT01  = 000002          1866#  1876
BIT02  = 000004          1865#  1875
BIT03  = 000010          1864#  1874
BIT04  = 000020          1863#  1873
BIT05  = 000040          1862#  1872   3389
BIT06  = 000100          1861#  1871   3389
BIT07  = 000200          1860#  1870   3389
BIT08  = 000400          1859#  1869   3390  11957
BIT09  = 001000          1858#  1868   3368   3389  11965  12362
BIT1   = 000002          1876#  10247
BIT10  = 002000          1857#  7517  12347
BIT11  = 004000          1856#  7517  11972
BIT12  = 010000          1855#  7517
BIT13  = 020000          1854#  7517  10263  10265  12354
BIT14  = 040000          1853#  7515  10240  11943
BIT15  = 100000          1852#  10235  10242  10249  10258  10260
BIT2   = 000004          1875#
BIT3   = 000010          1874#
BIT4   = 000020          1873#
BIT5   = 000040          1872#
BIT6   = 000100          1871#
BIT7   = 000200          1870#  7517
BIT8   = 000400          1869#  7517
BIT9   = 001000          1868#  7517
BLT1     037564          9810#  9815
BLT2     037606          9810   9811   9817#
BLT3     037616          9818   9819#
BPTVEC=  000014          1884#
CA       001712          2727#
CC       001736          2737#
CHECKD=  104411         12539#
```

```
CHECKE  040302        6287  6394  6512  6732  6854  6975  8054  9967#
CHECKT  040120        5106  6021  6267  6374  6492  6606  6711  6833  6954  7062  7154  7662  7775
                      8033  8155  9076  9180  9331  9439  9547  9920# 10095 10185 10347 12539
CLAREA  040002        6291  6398  6402  6516  6520  7029  7050  7118  7123  7170  7175  7249  7261
                      7268  7280  7315  7327  7392  7412  7457  7626  7631  7676  7742  7857  7877
                      7936  8127  8145  8455  8470  8540  8555  8628  8637  8709  8718  8771  8786
                      6839  8854  8910  8919  8975  8984  9043  9064  9149  9168  9867#
CLDISK  040064        3134  3178  3349  3547  3606  3691  3956  4017  4081  4143  4181  4201  4263
                      4300  4349  4385  4444  4516  4620  4701  4758  4844  5127  5250  5367  5495
                      5521  5641  5777  5912  5977  6003  6205  6237  6341  6457  6576  6699  6821
                      6942  7054  7143  7509  7576  7651  7763  7842  8021  8149  8229  8244  8335
                      8407  8417  8501  8584  866R  9015  9068  9148  9172  9276  9304  9391  9412
                      9503  9525  9611  9903# 10084 10173 10344 10529 10943
CLR     = 000040      2486#  3709  3723  5200  5308  5433  5578  5713  5850  7528  8338  9308
COMHD   044302        6715  6837  6958  7065  7156  7665  7778  8037  8158  9079  9191 10098 10187
                     10365 11014#
COMPA   045324       11146* 11189 11261# 11404*
COMPAR  040736        3647  4478  4584  4665  4738  4806  5180  5230  5291  5336  5416  5459  5560
                      5605  5695  5739  5832  5876  5952  6186  6301  6412  6530  6748  6870  6991
                      7335  7464  7943  8070  8391  9223  9374  9486  9594 10111# 10385 10411
COMWHD  047372        6274  6382  6499 11500#
COUNTD  045772       11398# 11402* 11417
CR      = 000015      1792# 12117 12127
CRC     041242        6248  6351  6468  6587  6693  6815  6936  7042  7137  7645  7757  8015  8139
                      9056  9162 10078 10167 10218# 10321
CRLF    = 000200      1793# 12088 12127
CSF     = 000002      2607#
CSU     = 000010      2609#
CS1     001700        2722#  3969  3981  6619  8295  9923  9926  9932  9970  9977 13230 13233 13239
                     13245 13252 13261 13264 13268
CS2     001676        2719#  3961  3990  4021  4036  4091  4152  4185  4205  7293  7427  7429  7555
                      7895  7897 13230 13233 13239 13252
CYL     044412        6685*  6694  6807*  6816  6928*  6937  7034*  7043  7130*  7138  7638*  7646  7750*
                      7758  8007*  8016  8132*  9048*  9057  9154*  9163 10071* 10079 10159* 10168 10315*
                     10322 11034#
DAREAD  044466       11076 11086#
DATENV  042540       10606# 11289* 11306* 11355* 11370* 13280
DAWORD  044472        6691*  6813*  6934*  7135*  7643*  7755*  8013*  10076*  10164*  10320*  11087#
DB      001670        2716# 13239
DCK     = 100000      2556#  8091 10440 10445 10461 10464
DCL     = 000100      2649#
DCLEAR  002024        2796#  4880
DDISP   = 177570      1799#  1979  2859
DENVL   = 000200      2566#
DE1     = 000040      2528#
DFF20   = 000002      2524#
DF1     063320        2036  2155  2162 13284#
DF11    063336        2144  2453 13290#
DF14    063345        2175 13293#
DF15    063355        2184 13296#
DF16    063357        2195 13297#
DF17    063363        2208 13299#
DF2     063325       13286#
DF20    063372        2221 13302#
DF21    063401        2232 13305#
DF22    063406        2249  2263 13307#
```

```
DF24    063412          2278    2291   13309#
DF26    063420          2304   13311#
DF27    063427          2319   13314#
DF3     063331          2124    2131   13288#
DF30    063435          2332   13316#
DF31    063442          2087    2095   13318#
DF32    063452          2071    2112   13321#
DF33    063463          2054   13324#
DF34    063473          2350   13327#
DF35    063502          2369    2441   13330#
DF36    063512          2384    2401   13333#
DF37    063520          2424   13335#
DF5   = 000001          2523#
DH1     056477          2031    2150   12853#
DH11    056745          2136    2445   12882#
DH14    057124          2167   12901#
DH15    057326          2181   12923#
DH16    057350          2189   12926#
DH17    057467          2200   12940#
DH2     056622          2118   12868#
DH20    057652          2213   12960#
DH21    060026          2226   12978#
DH22    060147          2243   12992#
DH24    060246          2271    2284   13003#
DH26    060404          2295   13019#
DH27    060566          2311   13039#
DH30    060725          2325   13056#
DH31    061036          2077   13069#
DH32    061240          2060    2101   13091#
DH33    061463          2045   13117#
DH34    061674          2340   13141#
DH35    062047          2360    2431   13159#
DH36    062243          2375    2392   13181#
DH37    062401          2412   13198#
DIGB  = 000004          2525#
DISK    046330          3613    3620    3649    6303    6414    6532    6670    6792    6913    7030    7077    7119    7120
                        7269    7270    7274    7281    7282    7400    7406    7413    7414    7627    7628    7743    7744
                        7865    7871    7878    7879    7883*   7992    8128    8168    9044    9089    9150    9244   10302
                       10413   10850   10864*  10865*  11423   11482#  11884   13271
DISPLA  001142          1979#   2859*   2867*  11986*  12346*
DISPRE  000174          1900#   2867
DLT   = 100000          2496#   3959    4152    4235    4240
DL64  = 000020          2527#
DMD   = 000001          2560#   3353    3356    3377    3704    4446    4529    4632    4702    4759    4857    4879    5129
                        5202    5252    5310    5369    5435    5498    5516    5517    5523    5580    5643    5668    5669
                        5715    5779    5805    5806    5852    5914    5929    5933    5978    6034    6043    6165    6206
                        6613    8230    8245    8364    8368    8418    8502    8585    8669    9016    9205    9277    9307
                        9349    9357    9415    9462    9469    9528    9570    9577   10495   10523   10789   11022   11150
                       11155   11269   11282   11285   11299   11302   11342   11352   11367   11508   11637   11640   11648
                       11658   11796   11805
DPR   = 000400          2531#   7541    7560    7588    8268    9111    9114    9944    9946    9989
DRY   = 000200          2530#   4715    4783    5159    5209    5271    5317    5386    5442    5541    5588    5677    5722
                        5814    5859    7541    7560    7588    8268    8370    9111    9114    9944    9951    9984
DST     001704          2724#   6051*   6122*  13249
DSWR  = 177570          1798#   1978    2858
DS1     001722          2731#   7539    7558    7586    8266    9109    9115    9942    9946    9951    9984    9989   13230
```

```
                    13233   13242   13245   13252   13261   13264   13268
DT          001724   2732#
DTAGAP      047334  11485#
DTE     =   010000   2553#
DTSY    =   001000   2568#  11171   11234   11237
DT1         062620   2035    2154    2161   13224#
DT11        062662   2143    2452   13230#
DT14        062702   2174   13233#
DT15        062724   2183   13236#
DT16        062732   2194   13237#
DT17        062746   2207   13239#
DT2         062634  13226#
DT20        062766   2220   13242#
DT21        063006   2231   13245#
DT22        063022   2248    2262   13247#
DT24        063034   2277    2290   13249#
DT26        063054   2303   13252#
DT27        063072   2318   13255#
DT3         062646   2123    2130   13228#
DT30        063110   2331   13258#
DT31        063124   2086    2094   13261#
DT32        063146   2070    2111   13264#
DT33        063172   2053   13268#
DT34        063214   2348   13271#
DT35        063234   2367    2439   13274#
DT36        063256   2382    2399   13277#
DT37        063274   2422   13280#
DVA     =   004000   2516#   3558    3561    3969    3981    4269    4271    4278    4282    4548    7784    7786    9841
                     9923    9926    9977
ECDATA      042520  10587#  10635   10790*  10856*  10858*  11279*  11281*  11296*  11298*  11345*  11349*  11360*  11364*
ECH     =   000100   2547#
ECI     =   004000   2633#   4645    6705    6827    6948    7150    7658    7771    8027    10091   10181   10362
ECORR       043144  10784#
ECTEST      042556  10626#  10818   10859   11290   11307   11356   11371
EC1         001730   2734#  13274   13277   13280
EC2         001732   2735#  13274   13277   13280
EMTVEC=     000030   1887#   2843*   2844*   2938*   2965*
EM1         053706   2030    2309   12598#
EM11        054057   2135    2148    2293   12617#
EM14        054112   2166   12622#
EM15        054132   2179   12625#
EM16        054203   2188   12632#
EM17        054224   2199   12635#
EM2         053771   2043    2058    2075   12607#
EM20        054250   2212   12639#
EM21        054271   2225   12642#
EM22        054313   2236   12646#
EM24        054706   2268   12693#
EM25        055001   2281   12703#
EM30        055041   2324   12709#
EM31        055163   2337   12724#
EM32        055306   2354   12739#
EM33        055572   2374   12772#
EM34        055644   2387   12779#
EM35        055737   2406   12789#
EM36        056236   2429   12824#
```

```
EM37    056344          2444   12836#
EM6     054020          2099    2117   12611#
ERCLFC  016712          4893    4904    4915   4928   4947   4957   4967   4977   4987   4997   5007   5017   5026
                        5037    5046    5057   5066   5077   5096#
ERCRC   047536         11519*  11561#  11721*
ERCS2C  012252          3736    3747    3761   3773   3783   3793   3803   3813   3823   3833   3842   3852   3863
                        3874    3886    3899   3910   3930#
ERFLG$  001772          2752#   3610*   4095*  4109   6271*  6283   6297*  6379*  6390   6408*  6496*  6508   6526*
                        6610*   6714*   6728   6836*  6850   6957*  6971   7064*  7072   7085   7155*  7166   7191
                        7333*   7443*   7664*  7675*  7695   7777*  7884*  7922*  8036*  8050   8157*  8165   8176
                        9078*   9086    9097   9182*  9239*  9252   9336*  9444*  9552*  10132  10364*  10374  11073
                       11591   12340*
ERHDGP  047540         11520*  11568#  11731*
ERHEAD  047534         11518*  11549#  11710*
ERPOS   043142         10780#  10786*  10803
ERR     = 040000        2537#   7541    7560   7588   8268   8371   9206   9358   9470   9578
ERRVEC= 000004          1880#   2856    2857*  2868*  2940*  2947*  2967*  11948  11949*  11951*  11954*
ERSTAR  044256         10943#
ERUNIT  044254         10942#  10944
ERWORD  044532          4485    4592    4673   4746   4814   5188   5239   5298   5345   5423   5468   5568   5614
                        5703    5748    5840   5885   5960   6194   7079*  7084*  7185*  7190*  7689*  7694*  8170*
                        8175*   8398    9091*  9096*  9231   9246*  9251*  9382   9494   9602   10125*  10131*  11137#
                       11173*  11193*  11197*  11219*  11239*  11666*  11672*  11676*  11694*  11701*  11707*  11720*  11724*
                       11728*  11739*  13226   13228   13261   13264   13268
ER1     001702          2723#   7596    8091   8195   8249   8259   13230  13233  13242  13252  13261  13264  13268
                       13274
ER2     001706          2725#  13242
ER3     001714          2728#  13242
EXT1    = 000001        2587#
EXT10 = 000010          2590#
EXT2    = 000002        2588#
EXT20 = 000020          2591#
EXT4    = 000004        2589#
EXT40 = 000040          2592#
FEN     = 000200        2613#
FER     = 000020        2545#   8091    8195
FILLEC  043316         10840#
FIRST   004174          2816#   2875    2912*
FMT22 = 010000          2634#   4645    6009   6242   6258   6272   6345   6361   6380   6462   6479   6497   6581
                        6597    6611    6685   6705   6734   6807   6827   6856   6928   6948   6977   7034   7060
                        7130    7150    7245   7310   7638   7658   7750   7771   8027   8153   8349   8463   8548
                        8779    8847    9048   9074   9154   9178   9318   9337   9426   9445   9553   10071   10091
                       10159   10181   10315  10362  10489
FNWORD  047556          6247*   6350*   6467*  6586*  11594#
FORMAT  045774         11399#  11416
FOUT    047562         11592   11596#
FRMAT1  031424          8127#
FSYNER  047532         11517*  11537#  11697*
FUTABL  002016          2792#   8232
GCRC    047526          6250    6353    6470   6589   8141  11527#  11716  11719
GECC1   042522         10591#  10633   10716*  10848*  10864   13271   13274   13280
GECC2   042524         10594#  10634   10717*  10849*  10865   13271   13274   13280
GNS     = ****** U      1899    2882    2886   2890   2894   2898   2902   2906   2910   2919   2955   2959   3025
                        3048    3052    3056   3060   3065   3091   3097   3103   3140   3147   3154   3185   3189
                        3193    3199    5394   9632   9638   9727   9733   9740   9744   9748   9755   9759  10885
                       10891   10902   10908  10914  10918  10924  10930  11907  12528  12529  12530  12531  12532
```

```
                    12535   12536   12537   12538   12539   12540
GO      = 000001     2510#    4457    4641    4712    4780    4881    5153    5157    5207    5265    5269    5315    5382
                     5384    5440    5536    5539    5585    5663    5675    5720    5800    5812    5857    5928    6035
                     6614    6616    7549    8257    8372    9350    9463    9571   10496   10524   11023   11509
GRV     = 000010     2526#
HADTMP  042544      10616#  10805
HARDER  042536      10603#
HCCRCE  041644       8480    8565    8647    8728    8796    8864    8929    8994   10342#
HCE     = 000200     2548#  10430   10434   10440   10445
HCI     = 002000     2632#    4645
HCRC    = 000400     2549#  10452   10455   10461   10464
HDESYN  047542      11521*  11579#  11742*
HDWSYN  046326      11480#
HEADER  046302      11477#
HEDGAP  044426      11031*  11050#  11200*
HEDSYN  044430      11032*  11061#  11220*  11241*
HEGAP   046314      11191   11479#
HT      = 000011     1790#  12086   12127
IAE     = 002000     2551#   9354    9466    9574
IE      = 000100     2511#
ILF     = 000001     2541#   8259
ILLEGL  002060       2810#   8242*   8246
ILR     = 000002     2542#
IOTVEC= 000020       1885#   2841*   2842*
IR      = 000100     2487#   3568    3959    3989    4019    4034    4183    4203    4240    4306    4312    4322    4546
                     7291    7425    7792    9198    9846
IXE     = 004000     2617#
KEY1    044416       6689*   6811*   6932*   7038*   7133*   7641*   7753*   8011*   8135*   9052*   9158*  10074*  10162*
                    10318*  11036#
KEY2    044420       6690*   6812*   6933*   7039*   7134*   7642*   7754*   8012*   8136*   9053*   9159*  10075*  10163*
                    10319*  11037#
KIPAR0= 172340       1936#
KIPAR1= 172342       1937#
KIPAR2= 172344       1938#
KIPAR3= 172346       1939#
KIPAR4= 172350       1940#
KIPAR5= 172352       1941#
KIPAR6= 172354       1942#
KIPAR7= 172356       1943#
KIPDR0= 172300       1925#
KIPDR1= 172302       1926#
KIPDR2= 172304       1927#
KIPDR3= 172306       1928#
KIPDR4= 172310       1929#
KIPDR5= 172312       1930#
KIPDR6= 172314       1931#
KIPDR7= 172316       1932#
LA      001734       2736#
LAD     040044       9818*   9885#   9889
LERR    037532       9801#   9807*   9848
LF      = 000012     1791#  12121   12127
LST     = 002000     2533#   9111    9114    9206
MAKECY  042400       5979    6207    8419    8503    8586    8670    9017    9278   10515#
MASK    037530       9800#   9804*   9828*   9829    9830*   9833*   9834    9835*
MCLK    = 000002     2561#   6074    6075    6076    6077    6103    6104    6105    6106    6152    6153    6156    6157
                    10816   10817   10823   10824   11152   11153   11155   11156   11158   11159   11274   11291   11342
```

```
                    11357  11640  11650  11823  11825  11826  11827  11828  11842  11843  11844  11845
MCPE  = 020000       2517#
MHS   = 001000       2615#
MID     042370      10491*  10498#
MIDDLE  042266       8282   10480#
MINX  = 000004       2562#   5502   5503   5648   5649   5785   5786   6042   9308   9309   9416   9417   9529
                     9530   11804
MMVEC = 000250       1914#
MOL   = 010000       2535#   3179   3195
MPE   = 000400       2489#
MR      001720       2730#  13277
MRC   = 000020       2564#  11272  11277  11285  11294  11302
MSE   = 000020       2610#
MSTCK = 000010       2563#   5516   5668   5805   6036   6037   6039   6040   6042   6074   6075   6103   6104
                     6138    6139   6141   6142   6156   6157  11151  11153  11155  11156  11274  11342  11639
                    11640   11797  11798  11799  11800  11804  11822  11825  11826  11842  11843  11852  11853
MWR   = 000040       2565#  11343  11358  11641  11651
MXF   = 001000       2490#   8292
NCODE   042530      10600#
NCOUNT  042532      10601#  10796*
NED   = 010000       2493#   3622   10946
NEM   = 004000       2492#   7792
NEXT1   011240       3653    3690#
NHS   = 002000       2616#
NOPERA  002016       2793#   4703   4771
NOSYNC  044526       8125*   8449*  8534*  8619*  8700*  8763*  8831*  8899*  8964*  11077  11134#  11208  11232
                    11932*
NOUNIT  001762       2743#   3068*  3078*  3110*  3115   9652*
NOWORD  044460       7040*   8137*  9054*  9160*  11083#
NUNIT   001764       2745#   3115*  3116*  4938
OCYL  = 100000       2651#
OF      001710       2726#
OFREV = 000200       2631#
OFSETC  002050       2806#   5527
OF100 = 000004       2626#
OF200 = 000010       2627#
OF25  = 000001       2624#   5528   5586
OF400 = 000020       2628#
OF50  = 000002       2625#
OF800 = 000040       2629#
OPERSE  036710       9723#  12169
OPI   = 020000       2554#
OR    = 000200       2488#   4032   4034   4089   4200   4216   4240   9197   9846
OUT     044476      11074   11081  11085  11089#
PAR   = 000010       2544#   3578   3581
PAT   = 000020       2485#   3550   3568
PC    =%000007       1811#   2874*  2992*  3020*  3134*  3178*  3225*  3251*  3277*  3304*  3330*  3349*  3418*
                     3444*   3470*  3496*  3522*  3547*  3606*  3655*  3657*  3691*  3736*  3747*  3761*  3773*
                     3783*   3793*  3803*  3813*  3823*  3833*  3842*  3852*  3863*  3874*  3886*  3899*  3910*
                     3935*   3956*  3960*  3980*  4017*  4020*  4035*  4081*  4090*  4143*  4150*  4181*  4184*
                     4201*   4204*  4263*  4300*  4349*  4385*  4444*  4492*  4516*  4540*  4599*  4620*  4678*
                     4701*   4751*  4758*  4819*  4844*  4893*  4904*  4915*  4928*  4947*  4957*  4967*  4977*
                     4987*   4997*  5007*  5017*  5026*  5037*  5046*  5057*  5066*  5077*  5101*  5106*  5127*
                     5195*   5246*  5250*  5304*  5350*  5367*  5430*  5474*  5496*  5521*  5574*  5621*  5641*
                     5709*   5755*  5777*  5846*  5892*  5912*  5966*  5977*  6003*  6021*  6198*  6205*  6237*
                     6267*   6274*  6287*  6311*  6317*  6341*  6374*  6382*  6394*  6422*  6432*  6457*  6492*
```

```
                    6499*  6512*  6540*  6550*  6576*  6606*  6618*  6699*  6711*  6715*  6732*  6758*  6762*
                    6821*  6833*  6837*  6854*  6880*  6884*  6942*  6954*  6958*  6975*  7001*  7005*  7054*
                    7062*  7065*  7071*  7143*  7154*  7156*  7285*  7292*  7345*  7351*  7417*  7426*  7476*
                    7482*  7509*  7531*  7550*  7576*  7580*  7651*  7662*  7665*  7763*  7775*  7778*  7842*
                    7885*  7894*  7955*  7961*  8021*  8033*  8037*  8054*  8078*  8082*  8089*  8149*  8155*
                    8158*  8164*  8229*  8244*  8248*  8258*  8282*  8293*  8335*  8402*  8407*  8417*  8452*
                    8501*  8537*  8584*  8623*  8668*  8704*  8766*  8834*  8903*  8968*  9015*  9068*  9076*
                    9079*  9085*  9148*  9172*  9180*  9191*  9236*  9242*  9276*  9304*  9331*  9387*  9391*
                    9412*  9439*  9499*  9503*  9525*  9547*  9607*  9611*  9675*  9678*  9688*  9693   9790*
                    9810*  9811*  9825*  9848*  9850*  9911*  9922*  9960*  9969*  9993*  10084*  10095*  10098*
                   10101*  10134*  10136*  10173*  10185*  10187*  10190*  10327*  10344*  10347*  10365*  10373*  10393*
                   10401*  10419*  10424*  10501*  10529*  10737*  10743*  10770*  10811*  10818*  10859*  10872*  10943*
                   11096*  11166*  11170*  11178*  11180*  11182*  11184*  11186*  11194*  11205*  11227*  11231*  11290*
                   11307*  11313*  11356*  11371*  11381*  11424*  11430*  11436*  11440*  11443*  11603*  11673*  11689*
                   11702*  11714*  11725*  11735*  11886*  11894*  12095*  12102*  12109*  12123*  12125*  12153*  12167*
                   12356*  12423*  12580
PCJSR   002000      2765#   9920*   9921*   9967*   9968*  10342*  10343*  10517*  10518*  10784*  10785*  11014*  11015*
                   11500*  11501*  13252   13255   13258   13264   13268   13277
PGE   = 002000      2491#   6619
PIE1  = 100000     10554#  10701   10715
PIE10 = 000100     10563#
PIE11 = 000040     10564#  10668
PIE12 = 000020     10565#  10701
PIE13 = 000010     10566#
PIE14 = 000004     10567#
PIE15 = 000002     10568#
PIE16 = 000001     10569#
PIE17 = 100000     10570#
PIE18 = 040000     10571#
PIE19 = 020000     10572#
PIE2  = 040000     10555#  10659
PIE20 = 010000     10573#
PIE21 = 004000     10574#  10676
PIE22 = 002000     10575#  10707
PIE23 = 001000     10576#  10684
PIE24 = 000400     10577#  10707
PIE25 = 000200     10578#
PIE26 = 000100     10579#
PIE27 = 000040     10580#
PIE28 = 000020     10581#
PIE29 = 000010     10582#
PIE3  = 020000     10556#  10701
PIE30 = 000004     10583#
PIE31 = 000002     10584#
PIE32 = 000001     10585#  10640   10648
PIE4  = 010000     10557#
PIE5  = 004000     10558#
PIE6  = 002000     10559#
PIE7  = 001000     10560#
PIE8  = 000400     10561#
PIE9  = 000200     10562#
PIP   = 020000      2536#   5158    5208    5270    5316    5385    5441    5540    5587    5676    5721    5813    5858
PIRQ  = 177772      1797#
PIRQVE= 000240      1891#
PKACK   002054      2808#   4448
PLU   = 020000      2619#
```

DERPSB.P11 CROSS REFERENCE TABLE -- USER SYMBOLS

```
POSITI  042534          10602#  10740   10794   10802*  10803   10805   10847*  13274   13280
PRBA    010006           3245*   3246#
PRCA    010570           3490*   3491#
PRCS1   010116           3298*   3299#
PRCS2   010052           2986    3271*   3272#
PRDST   010414           3412*   3413#
PRE   = 000020           2647#
PRER1   010162           3324*   3325#
PRER2   010460           3438*   3439#
PRER3   010634           3516*   3517#
PROF    010524           3464*   3465#
PRCG  = 001000           2532#   3860    4557    5034    7540    7559    7587    8267    9110    9943
PRWC    007742           3219*   3220#
PR0   = 000000           1814#
PR1   = 000040           1815#
PR2   = 000100           1816#
PR3   = 000140           1817#
PR4   = 000200           1818#
PR5   = 000240           1819#
PR6   = 000300           1820#
PR7   = 000340           1821#
PS    = 177776           1794#   1795    2872*   9629*   9724*
PSEL  = 000000           2515#
PSU   = 000001           2644#
PSW   = 177776           1795#
PUTREG  037470           3960    3980    4020    4035    4090    4150    4184    4204    6618    7071    7292    7426    7531
                         7550    7580    7894    8089    8164    8248    8258    8293    9085    9242    9777#   9922    9969
                        10373   10737   10743   10811
PWRVEC= 000024            1886#   2847*   2848*  12549*  12549*  12558*  12564*  12576*  12577*
P12     042550          10618#  10670*  10671*  10700
P22     042552          10619#  10678*  10679*  10705
P24     042554          10620#  10686*  10687*  10706
P3      042546          10617#  10661*  10662*  10699
RCOM    050700          11872#  11877*
RCYL    044516          11104#  11142*  11177
RDCHR = 104405          12240   12535#
RDHEAD  044534          11033   11142#
RDLIN = 104406          12290   12536#
RDOCT = 104407           2921    9750    9761   10893   10910   12537#
RDY   = 000200           2512#   3558    3561    3969    3981    4269    4271    4278    4282    4548    4719    4787    7784
                         7786    8373    9124    9841    9923    9932    9970
READ    045326          11166   11170   11178   11180   11182   11184   11186   11227   11231   11266#  11886   11894
READAT  002042           2803#   7144    7652    7764
READIN  002056           2809#   4630
RECALI  002022           2795#   5791
REDATA  050702          11086   11876#
REFOR   002044           2804#   6713    6835    6956    8035    9334
REGADR  037534           3010*   3361*   3383*   3562*   3573*   3582*   3716*   3930*   4049*   4058*   4108*   4162*   4220*
                         4228*   4243*   4272*   4283*   4313*   4330*   4359*   4369*   4395*   4405*   4487*   4594*   4675*
                         4748*   4816*   5096*   5190*   5241*   5300*   5347*   5425*   5470*   5570*   5616*   5705*   5750*
                         5842*   5887*   5962*   6196*   7553*   7787*   7797*   7805*   8400*   9113*   9233*   9384*   9496*
                         9604*   9802#   9806*  10433*  10444*  10454*  10465*  10534*  13224   13255   13258
REGSA1  052716           2938   12341#
REINTO  003126           2814#   4453    4460*   4473    4479    4535    4544*   4547*   4548*   4549*   4550*   4551*   4552*
                         4553*   4554*   4556    4561*   4562*   4563*   4574    4580    4585    4637    4644*   4645*   4647*
                         4648*   4660    4666    5149    5157*   5158*   5159*   5175    5181    5207*   5208*   5209*   5210*
```

```
                    5211*  5226   5231   5261   5269*  5270*  5271*  5286   5292   5315*  5316*  5317*  5318*
                    5319*  5332   5337   5378   5384*  5385*  5386*  5411   5417   5440*  5441*  5442*  5443*
                    5456   5460   5532   5539*  5540*  5541*  5555   5561   5585*  5586*  5587*  5588*  5589*
                    5601   5606   5659   5675*  5676*  5677*  5690   5696   5720*  5721*  5722*  5723*  5735
                    5740   5796   5812*  5813*  5814*  5827   5833   5857*  5858*  5859*  5860*  5872   5877
                    5924   5933*  5934*  5947   5953   6016   6162*  6163*  6165*  6166*  6167*  6181   6187
                    6292   6293   6302   6399   6400   6403   6404   6413   6517   6518   6521   6522   6531
                    6701   6750   6823   6872   6944   6993   7124   7125   7146   7183   7244   7250   7251
                    7255   7262   7263   7337   7381   7386   7393   7394   7466   7510   7525   7632   7633
                    7654   7687   7846   7851   7858   7859   7914   7945   8023   8072   8356   8368*  8369*
                    8370*  8371*  8372*  8373*  8386   8392   8456   8457   8486   8541   8542   8571   8638
                    8639   8719   8720   8772   8773   8802   8840   8841   8870   8920   8921   8985   8986
                    9187   9194*  9195*  9197*  9198*  9199*  9200*  9201*  9203*  9204*  9205*  9206*  9218
                    9224   9315   9343   9353*  9354*  9355*  9356*  9357*  9358*  9369   9375   9456   9466*
                    9467*  9468*  9469*  9470*  9481   9487   9564   9574*  9575*  9576*  9577*  9578*  9589
                    9595   10087  10177  10387  10412
RELEAS  002026      2797#  5919
RESVEC= 000010      1881#
RETCL   002052      2807#  5654
RHAS    001646      2701#  3002   3027   7520   7579*
RHBA    001624      2686#  3245   3695*  4386   4520*  4625*  4848*  6256*  6359*  6477*  6595*  6701*  6823*
                    6944*  7056*  7146*  7654*  7766*  7800   7805   8023*  8151*  8346*  9070*  9174*  9316*
                    9424*  9537*  10087* 10177* 10359* 10486*
RHCA    001642      2699#  3490   3702*  4527*  4628*  4855*  5143*  5257*  5510*  6008*  6273*  6381*  6498*
                    6612*  6709*  6831*  6952*  7061*  7153*  7661*  7774*  8031*  8154*  8336   8350*  9075*
                    9179*  9338*  9446*  9554*  10094* 10184* 10351* 10490* 10520*
RHCC    001666      2709#  5137   5510   9783   10530  10534
RHCS1   001630      2694#  3298   3629   3697*  4448*  4457*  4522*  4631*  4641*  4850*  5136*  5153*  5256*
                    5265*  5373*  5382*  5527*  5536*  5654*  5663*  5791*  5800*  5919*  5928*  6035*  6614*
                    6616*  7549*  8343*  8355   8365*  8379   8400   9121   9350*  9463*  9571*  9839   9903
                    10483* 10496* 10522* 10524* 10887  10920  11023* 11509*
RHCS2   001626      2687#  2987   3028   3084*  3220*  3246*  3271   3272*  3299*  3325*  3413*  3439*  3465*
                    3491*  3517*  3550*  3696*  4029   4197   4521*  4541*  4767*  4849*  6621   7528*  7529*
                    7663*  7776*  8338*  8339*  8348*  9824*  9843   9904   10488*
RHDB    001620      2684#  2942   3693*  3711   3957   4027*  4028*  4042   4043   4049   4058   4084*  4098
                    4108   4146*  4156   4162   4190   4210   4846*  4882   10894
RHDST   001634      2696#  3412   3699*  4524*  4626*  4852*  5135*  6006*  6031   6051   6122   6263*  6370*
                    6488*  6602*  6704*  6826*  6947*  7059*  7149*  7657*  7770*  8026*  8152*  8347*  9073*
                    9177*  9201   9327*  9435*  9543*  10090* 10180* 10356* 10487* 10521*
RHDS1   001652      2703#  4710   4778   8275   9113   9905
RHDT    001654      2704#  3085   3087   3099   3156   3160   3162   3635   3637
RHEC1   001660      2706#  10740
RHEC2   001662      2707#  4706   4765   4774   10734
RHER1   001632      2695#  3031   3324   3552*  3698*  4523*  4851*  7512   9455   9474   9496   9563   9582
                    9604   9906   10429  10433  10439  10444  10451  10454  10459  10465
RHER2   001636      2697#  3438   3700*  4525*  4853*  7514
RHER3   001644      2700#  3516   3703*  4528*  4856*  7516
RHLA    001664      2708#  5210   5318   5443   5589   5723   5860   5934   6050   6120
RHMR    001650      2702#  3350   3704*  3705*  4446*  4529*  4530*  4632*  4702*  4759*  4857*  4858*  4879*
                    5129*  5202*  5252*  5310*  5369*  5435*  5498*  5502*  5503*  5516*  5517*  5523*  5580*
                    5643*  5648*  5649*  5668*  5669*  5715*  5779*  5785*  5786*  5805*  5806*  5852*  5914*
                    5929*  5978*  6030   6206*  6613*  8230*  8245*  8364*  8418*  8502*  8585*  8669*  9016*
                    9277*  9307*  9308*  9309*  9348   9415*  9416*  9417*  9461   9528*  9529*  9530*  9569
                    10495* 10523* 10788  11022* 11148  11415  11508* 11636  11794
RHOF    001640      2698#  3464   3701*  4526*  4627*  4854*  5528*  6009*  6272*  6380*  6497*  6611*  6705*
                    6827*  6948*  7060*  7150*  7658*  7771*  8027*  8153*  8349*  9074*  9178*  9337*  9445*
```

```
                    9553*  10091*  10181*  10362*  10489*
RHSN    001656      2705*   3149    3159    3161
RHWC    001622      2685*   3219    3694*   4350    4452    4466    4487    4519*   4534    4567    4594    4624*   4636
                    4653    4675    4704    4706    4726    4748    4764    4765    4772    4774    4794    4816    4847*
                    5148    5166    5190    5218    5241    5260    5278    5300    5325    5347    5377    5403    5425
                    5448    5470    5531    5547    5570    5594    5616    5658    5683    5705    5728    5750    5795
                    5820    5842    5865    5887    5923    5940    5962    6015    6174    6196    6254*   6357*   6474*
                    6593*   6700*   6822*   6943*   7055*   7145*   7653*   7765*   8022*   8150*   8345*   9069*   9173*
                    9186    9211    9233    9314*   9342    9362    9384    9422*   9535*   9781    9783   10086*  10176*
                   10357*  10485*
RKEY1   044522     11106#  11144*  11181
RKEY2   044524     11107#  11145*  11183
RMR   = 000004      2543#   8369
RNO     050676     11871#  11876*
RPVEC   001616      2674#   2873*  10904   10911*  10926   11910*
RPVECT  051012      2873   11904#  11910
RSETR   044520     11105#  11143*  11179
RSYNC   044514     11103#  11169   11174   11206   11217   11230   11240   11692   11695   11737   11741
RO    =%000000      1802#   2941*   2944*   3030*   3034*   3071*   3074*   3079*   3112*   3136*   3137    3350*   3353*
                    3354*   3358    3360    3361    3377*   3378*   3379    3382    3383    3614*   3616*   3619*   3624*
                    3641    3643*   3647*   3711*   3716    3717    3733    3744    3758    3770    3780    3790    3800
                    3810    3820    3830    3838    3849    3858    3871    3883    3896    3907    3916    3921    3930
                    3934    3957*   4042*   4044    4048    4076*   4078*   4082*   4084    4085*   4096*   4098*   4101*
                    4104    4106    4120    4144*   4145*   4146    4147    4190*   4192*   4210*   4211*   4213    4214
                    4220    4228    4451*   4465*   4478*   4533*   4566*   4584*   4635*   4652*   4665*   4704*   4707
                    4725*   4738*   4764*   4766*   4772*   4775    4793*   4806*   4864*   4867*   4882*   4890    4901
                    4912    4925    4943    4954    4964    4974    4984    4994    5004    5014    5023    5032    5043
                    5054    5063    5074    5083    5088    5096    5100    5103    5147*   5165*   5180*   5217*   5230*
                    5259*   5277*   5291*   5324*   5336*   5376*   5402*   5416*   5447*   5459*   5515*   5518*   5530*
                    5546*   5560*   5593*   5605*   5657*   5667*   5670*   5682*   5695*   5727*   5739*   5794*   5804*
                    5807*   5819*   5832*   5864*   5876*   5922*   5939*   5952*   5979*   6013*   6030*   6034*   6036*
                    6037*   6039*   6040*   6042*   6043*   6074*   6075*   6076*   6077*   6103*   6104*   6105*   6106*
                    6138*   6139*   6141*   6142*   6152*   6153*   6156*   6157*   6173*   6186*   6207*   6232*   6234*
                    6255*   6256    6258*   6260*   6291*   6301*   6336*   6338*   6358*   6359    6361*   6363*   6364*
                    6365*   6367*   6398*   6402*   6412*   6452*   6454*   6475*   6477    6479*   6481*   6482*   6483*
                    6485*   6516*   6520*   6530*   6571*   6573*   6594*   6595    6597*   6599*   6670*   6671*   6678*
                    6733*   6734*   6737*   6738*   6739*   6742*   6748*   6792*   6793*   6800*   6855*   6856*   6859*
                    6860*   6861*   6864*   6870*   6913*   6914*   6921*   6976*   6977*   6980*   6981*   6982*   6985*
                    6991*   7029*   7050*   7076*   7080    7082    7118*   7123*   7170*   7175*   7182*   7186    7188
                    7206    7249*   7254*   7256    7258*   7261*   7268*   7273*   7275    7277*   7280*   7309*   7310*
                    7311*   7312*   7313*   7315*   7320*   7322    7324*   7327*   7335*   7380*   7382    7383*   7385*
                    7387    7389*   7392*   7399*   7401    7402*   7405*   7407    7409*   7412*   7444*   7446    7447*
                    7450*   7452    7454*   7457*   7464*   7510*   7512*   7513*   7514*   7515*   7516*   7517*   7525*
                    7526    7566    7569    7626*   7631*   7676*   7686*   7690    7692    7710    7742*   7845*   7847
                    7848*   7850*   7852    7854*   7857*   7864*   7866    7867*   7870*   7872    7874*   7877*   7923*
                    7925    7926*   7929*   7931    7933*   7936*   7943*   7992*   7993*   8000*   8055*   8056*   8059*
                    8060*   8061*   8064*   8070*   8127*   8145*   8167*   8171    8173    8232*   8234    8336*   8337
                    8354*   8378*   8391*   8419*   8455*   8462*   8463*   8464*   8465*   8466*   8470*   8480*   8503*
                    8540*   8547*   8548*   8549*   8550*   8551*   8555*   8565*   8586*   8628*   8637*   8647*   8670*
                    8709*   8718*   8728*   8771*   8778*   8779*   8780*   8781*   8782*   8786*   8796*   8839*   8846*
                    8847*   8848*   8849*   8850*   8854*   8864*   8910*   8919*   8929*   8975*   8984*   8994*   9017*
                    9043*   9064*   9088*   9092    9094    9149*   9168*   9185*   9210*   9223*   9243*   9247    9249
                    9278*   9315*   9316    9318*   9320*   9321*   9322*   9324*   9341*   9348*   9349*   9361*   9374*
                    9423*   9424    9426*   9428*   9429*   9430*   9432*   9454*   9461*   9462*   9473*   9486*   9536*
                    9537    9540*   9562*   9569*   9570*   9581*   9594*   9654*   9656    9685*   9688    9778    9781*
                    9784    9789*   9871    9872    9873    9883*  10005   10006*  10007   10009   10010   10011   10012*
```

```
                    10052   10053   10054   10061*  10117   10118   10119   10120   10121   10122*  10150*  10219   10220*
                    10228   10280*  10297   10300*  10303   10326*  10342   10349   10350   10351   10352   10353   10354
                    10355   10357   10359   10360   10385*  10411*  10425   10469*  10481   10500*  10517   10519   10539*
                    10627   10642*  10649*  10652*  10653*  10654   10699*  10700*  10702   10704*  10705*  10706*  10708
                    10769*  10784   10786   10828*  10841   10871*  10894*  10896*  10897*  11016   11095*  11148*  11150*
                    11151*  11152*  11153*  11155*  11156*  11158*  11159*  11171   11234   11237   11269*  11272*  11274*
                    11277   11286*  11291*  11294   11303*  11335   11340*  11342*  11343   11352*  11357*  11358   11367*
                    11380*  11405   11415*  11451*  11502   11602*  11636*  11637*  11639*  11640*  11641   11648*  11650*
                    11651   11658*  11788   11794*  11796*  11797*  11798*  11799*  11800*  11804*  11805*  11822*  11823*
                    11825*  11826*  11827*  11828*  11842*  11843*  11844*  11845*  11852*  11853*  11863*  12003   12013*
                    12017   12033   12034   12047*  12078   12079*  12080   12083*  12287   12291*  12292   12295   12315*
                    12318*  12379   12380*  12381*  12388*  12389*  12390*  12391*  12392*  12393   12398   12404   12406*
                    12407   12421*  12512   12513*  12514   12515*  12516*  12517*  12518*  12550   12575*
R1      =%000001     1803#   2942*   2943    2949    3002*   3003*   3004    3010    3027*   3036    3039    3080    3351*
                    3354    3355    3366*   3367*   3368*   3375*   3378    3379    3381    3388*   3389*   3390*   3557
                    3562    3613*   3615*   3620*   3642*   3972    3978*   3984    4191*   4192    4194*   4195    4212*
                    4216    4264*   4265    4268    4272    4276*   4277    4283    4703*   4711    4712*   4771*   4779
                    4780*   4880*   4881*   5511*   5512*   6010*   6100*   6111*   6134*   6233*   6235*   6269*   6337*
                    6339*   6377*   6453*   6455*   6494*   6572*   6574*   6608*   6713*   6740*   6743*   6835*   6862*
                    6865*   6956*   6983*   6986*   7063*   7077*   7080    7083    7090    7144*   7183*   7186    7189
                    7206    7244*   7245*   7246*   7247*   7248*   7255*   7256*   7274*   7275*   7321*   7322*   7381*
                    7382*   7386*   7387*   7400*   7401*   7406*   7407*   7445*   7446*   7451*   7452*   7527*   7530
                    7551    7554    7564*   7566    7652*   7687*   7690    7693    7710    7764*   7767*   7782    7787
                    7846*   7847*   7851*   7852*   7865*   7866*   7871*   7872*   7924*   7925*   7930*   7931*   8035*
                    8062*   8065*   8156*   8168*   8171    8174    8185    8246*   8257*   8298    9077*   9089*   9092
                    9095    9102    9181*   9244*   9247    9250    9257    9334*   9442*   9550*   9655*   9656    9659
                    9779    9782*   9784*   9788*   9868    9871*   9874    9876*   9882*   9903*   9910*   9928    9934
                    9938    9973    9980    10049   10052*  10055   10060*  10096*  10112   10117*  10126   10129   10137
                    10149*  10186*  10221   10225*  10233   10240   10247   10255*  10258*  10260*  10263*  10265*  10268*
                    10270*  10275   10279*  10298   10301*  10304*  10306*  10310*  10325*  10350*  10482   10497*  10499*
                    10628   10633*  10639   10647   10675   10683   10698*  10707*  10708*  10714*  10716   10732   10768*
                    10787   10788*  10789*  10808   10816*  10817*  10823*  10824*  10827*  10842   10850*  10853   10870*
                    10895*  10898*  11017   11027*  11029*  11030   11031   11032   11033*  11082*  11086*  11094*  11142
                    11143   11144   11145   11146   11147   11243*  11244*  11402   11403   11404   11406   11410*  11413*
                    11435*  11438*  11442*  11445*  11450*  11452*  11503   11513*  11515*  11516   11517   11518   11519
                    11520   11521   11522*  11593*  11601*  11629   11630   11631   11632   11633   11634   11635*  11663*
                    11674*  11691*  11692   11696   11703*  11704   11709   11715*  11716   11718   11726*  11730   11736*
                    11737   11740   11744*  11745*  11787   11789   11839*  11850*  11862*  11864*  11876   11877   11878
                    11898*  11899*  12004   12017*  12018   12022   12046*  12288   12293*  12301*  12303*  12305*  12308*
                    12311   12314*  12403   12404*  12409   12413   12415   12420*  12551   12574*
R2      =%000002     1804#   2943*   3028*   3029*   3033*   3352*   3369*   3376*   3391*   3570    3573    3618*   3622
                    3626    3633*   3645*   3709*   3710*   3723*   3726*   3963    3994    4023    4038    4093    4154
                    4187    4207    4212    4215    4243    4301*   4302    4307    4313    4318*   4320*   4324    4330
                    4706*   4708*   4774*   4776*   4865*   4870*   4876*   5200*   5201*   5308*   5309*   5433*   5434*
                    5578*   5579*   5713*   5714*   5850*   5851*   6071*   6078*   6081*   6102*   6107*   6151*   6154*
                    6741*   6742    6863*   6864    6984*   6985    7078*   7084    7097*   7184*   7190    7213*   7295
                    7526*   7530*   7551    7553    7688*   7694    7717*   7794    7797    8063*   8064    8169*   8175
                    8190*   8289    9090*   9096    9107*   9245*   9251    9263*   9780    9783*   9785*   9787*   9869
                    9872*   9874*   9875*   9877*   9878*   9881*   9904*   9908*   9909*   10050   10053*  10055*  10059*
                    10113   10118*  10126   10130   10137   10148*  10222   10242*  10244*  10245*  10261   10278*  10299
                    10302*  10303*  10308*  10324*  10629   10634*  10658   10667   10697*  10701*  10702*  10713*  10715*
                    10717   10767*  10843   10851*  10862*  10869*  10944*  10946   11018   11093*  11157*  11160*  11164*
                    11167*  11192*  11197   11203*  11225*  11228*  11267   11273*  11308*  11312*  11336   11341*  11372*
                    11379*  11403*  11407   11426*  11449*  11504   11600*  11649*  11659*  11671*  11676   11681*  11699*
                    11707   11712*  11723*  11728   11733*  11790   11820*  11829*  11831*  11841*  11846*  11861*  11882*
                    11887*  11892*  11895*  12005   12016*  12020*  12023   12030*  12031*  12032   12037*  12045*  12289
```

```
                12294* 12302* 12304* 12306* 12312  12313* 12552  12573*
R3      =%000003 1805# 3072* 3073* 3076* 3106* 3179  3195  3629* 3630* 3631* 4214* 4225  4230
                 4705* 4707* 4773* 4775* 6031* 6032* 6033  6044  6056  6126  6144* 9809* 9813*
                 9814  9817* 9827  9870  9873* 9876  9880* 9905* 9948  9953  9956  9986  9991
                10051  10054* 10056* 10058* 10114 10119* 10123 10131 10142* 10147* 10223 10235* 10237*
                10238* 10245  10252  10256  10277* 10630  10639* 10640* 10641* 10642  10647* 10648* 10649
                10658* 10659* 10660* 10661  10667* 10668* 10669* 10670  10675* 10676* 10677* 10678  10683*
                10684* 10685* 10686  10766* 10844  10852* 10860* 10868* 11019  11092* 11337  11347* 11351*
                11362* 11366* 11376  11378* 11408  11411* 11412* 11423* 11425* 11431* 11437* 11441* 11444*
                11448* 11505  11599* 11644* 11647* 11654* 11657* 11663  11664  11668  11700* 11704  11708
                11791  11795* 11806  11854* 11860* 11884* 11885  11893  12006  12014* 12015* 12029* 12032*
                12041* 12042* 12044* 12236  12237* 12238  12241* 12242  12246  12248  12250* 12252* 12460
                12469* 12475* 12476* 12479* 12484* 12485* 12486  12495* 12553  12572*
R4      =%000004 1806# 3031* 3032* 3577  3582  3621  4215* 4235  4242  4350* 4351* 4352  4355
                 4359  4362* 4363  4369  4386* 4387* 4388  4391  4395  4398* 4399  4405  4869*
                 7520* 7578* 8251  8261  9805* 9806  9831* 9832  9838  9839  9843  9845  9906*
                10115  10123* 10124* 10125  10146* 10224 10249* 10251* 10252* 10266 10276* 10631 10765*
                10845  10853* 10854* 10867* 10945  11020 11091* 11409  11419* 11428 11432* 11447* 11506
                11598* 11792  11859* 12461  12463* 12464* 12465* 12466  12467* 12481 12483* 12491* 12494*
                12554  12571*
R5      =%000005 1807# 2988* 3077* 3084  3093  3106  3108  3111* 3221* 3247* 3273* 3300* 3326*
                 3414* 3440* 3466* 3492* 3518* 4043* 4052  4057  4077* 4079* 4083* 4086* 4097*
                 4099* 4102* 4127* 4213* 4218  4222  4485* 4486* 4487  4592* 4593* 4594  4673*
                 4674* 4675  4746* 4747* 4748  4765* 4769* 4814* 4815* 4816  4866* 4871* 5188*
                 5189* 5190  5239* 5240* 5241  5298* 5299* 5300  5345* 5346* 5347  5423* 5424*
                 5425  5468* 5469* 5470  5568* 5569* 5570  5614* 5615* 5616  5703* 5704* 5705
                 5748* 5749* 5750  5840* 5841* 5842  5885* 5886* 5887  5960* 5961* 5962  6072*
                 6082* 6098* 6133* 6134  6194* 6195* 6196  6248* 6257* 6261* 6351* 6366* 6368*
                 6468* 6484* 6486* 6587* 6596* 6600* 6669* 6672* 6675* 6679* 6693* 6791* 6794*
                 6797* 6801* 6815* 6912* 6915* 6918* 6922* 6936* 7042* 7137* 7521* 7532  7645*
                 7757* 7991* 7994* 7997* 8001* 8015* 8139* 8233* 8238* 8337* 8366* 8398* 8399*
                 8400  9056* 9162* 9231* 9232* 9233  9323* 9325* 9382* 9383* 9384  9431* 9433*
                 9494* 9495* 9496  9539* 9541* 9602* 9603* 9604  9804  9805  9807  9808* 9816*
                10078* 10116  10145* 10167* 10220 10275* 10281* 10321* 10516 10519* 10520 10531 10533
                10538* 10632  10764* 10846  10866* 11021 11090* 11149* 11162* 11191* 11202* 11222* 11268*
                11310* 11338  11339* 11374* 11377* 11507 11597* 11638* 11661* 11793  11821* 11832* 11840*
                11848* 11858* 12007  12009* 12011* 12018* 12022* 12037  12043* 12462  12468* 12470* 12472*
                12473* 12474* 12475  12493* 12555  12570*
R6      =%000006 1808# 1810  2835* 2836* 2837
R7      =%000007 1809# 1811
SAVDT   001774   2756# 3162* 3870  3872  5042  5044
SAVER   040542   4451  4465  4533  4566  4635  4652  4725  4793  5147  5165  5217  5259  5277
                 5324  5376  5402  5447  5530  5546  5593  5657  5682  5727  5794  5819  5864
                 5922  5939  6013  6173  8354  8378  9185  9210  9341  9361  9454  9473  9562
                 9581  10048#
SAVSN   001776   2759# 3161* 3882  3884  5053  5055
SC      = 100000 2519# 3558  3561  3969  6162  7784  7786  9199  9353
SCOP1   = 104410 12538#
SCRC    047610   11626# 11633#
SCYL    047600   11622# 11629* 11700
SC1     = 000100 2593#
SC10    = 001000 2596#
SC2     = 000200 2594#
SC20    = 002000 2597#
SC4     = 000400 2595#
SEARCH  050426   10497  11027  11513  11787#
```

```
SECGAP  046232          6232    6336    6452    6571   11475#  11635
SECOTR  044414          6687*   6688*   6809*   6810*   6930*   6931*   7036*   7037*   7131*   7132*   7639*   7640*   7751*
                        7752*   8009*   8010*   8133*   8134*   9050*   9051*   9156*   9157*  10072*  10073*  10160*  10161*
                       10316*  10317*  11024   11035#
SECTR   050424          6033*   6055    6125   11785#  11787*  11795
SEECOM  002046          2805#   5136    5256    5511   10522
SEGPER  047530         11516*  11529#  11669*  11679*
SELECT  001766          2747#   2827*   2829*   2913    3118    9643
SERCH   002030          2798#   6010
SETDSK  041516          8452    8537    8623    8704    8766    8834    8903    8968   10296#
SILOTB  046330          4076    4096    4101   11481#
SKEY1   047604         11624#  11631*
SKEY2   047606         11625#  11632*
SKI     = 040000        2650#
SN      001726          2733#
SND1    005306          2878    2912#
SP      =%000006        1810#   2839*   2856*   2864*   2868    2922*   2923    2924    2934*   2948    2961*   2981*   3093*
                        3099*   3108*   3142*   3149*   3156*   3213*   3239*   3265*   3292*   3318*   3355*   3356*   3357
                        3358    3406*   3432*   3458*   3484*   3510*   3543*   3567*   3568*   3569    3601*   3626*   3627*
                        3628*   3633    3640*   3641*   3642    3755*   3756*   3757    3858*   3859    3860*   3861    3951*
                        3958*   3959*   3961    3988*   3989*   3990    4012*   4018*   4019*   4021    4033*   4034*   4036
                        4088*   4089*   4091    4103*   4104    4107    4122*   4123*   4124    4126*   4129    4182*   4183*
                        4185    4202*   4203*   4205    4239*   4240*   4241    4262*   4265*   4266*   4267*   4302*   4303*
                        4304*   4305*   4306*   4308    4316*   4317*   4320    4321*   4322*   4323    4325    4348*   4352*
                        4353*   4354*   4381*   4388*   4389*   4390*   4439*   4511*   4545*   4546*   4547    4556*   4557*
                        4558*   4561    4615*   4629*   4630*   4631    4839*   4874*   4875*   4876    4922*   4923*   4924
                        5032*   5033    5034*   5035    5122*   5137*   5138    5363*   5396*   5491*   5636*   5772*   5907*
                        5976*   5999*   6057*   6058*   6059    6127*   6128*   6129    6204*   6227*   6331*   6447*   6566*
                        6658*   6668*   6671    6674    6702*   6703*   6704    6735*   6736*   6737    6780*   6790*   6793
                        6796    6824*   6825*   6826    6857*   6858*   6859    6901*   6911*   6914    6917    6945*   6946*
                        6947    6978*   6979*   6980    7024*   7057*   7058*   7059    7091*   7092*   7093    7113*   7147*
                        7148*   7149    7207*   7208*   7209    7237*   7290*   7291*   7293    7373*   7424*   7425*   7427
                        7504*   7539*   7540*   7541    7558*   7559*   7560    7586*   7587*   7588    7621*   7655*   7656*
                        7657    7711*   7712*   7713    7736*   7768*   7769*   7770    7791*   7792*   7793    7837*   7892*
                        7893*   7895    7980*   7990*   7993    7996    8024*   8025*   8026    8057*   8058*   8059    8119*
                        8186*   8187*   8188    8223*   8266*   8267*   8268    8330*   8416*   8443*   8500*   8528*   8583*
                        8613*   8667*   8694*   8757*   8825*   8893*   8958*   9014*   9038*   9071*   9072*   9073    9103*
                        9104*   9105    9109*   9110*   9111    9144*   9175*   9176*   9177    9258*   9259*   9260    9275*
                        9300*   9408*   9521*   9634*   9640*   9648*   9682*   9729*   9735*   9751*   9752    9762    9763*
                        9778*   9779*   9780*   9787    9788    9789    9845*   9846*   9847    9868*   9869*   9870*   9880
                        9881    9882    9889*   9920    9942*   9943*   9944    9967   10005*  10006   10011*  10012   10049*
                       10050*  10051*  10058   10059   10060   10088*  10089*  10090   10112*  10113*  10114*  10115*  10116*
                       10138*  10139*  10140   10145   10146   10147   10148   10149   10178*  10179*  10180   10219*  10221*
                       10222*  10223*  10224*  10276   10277   10278   10279   10280   10297*  10298*  10299*  10324   10325
                       10326   10352*  10354*  10356   10481*  10482*  10499   10500   10516*  10538   10627*  10628*  10629*
                       10630*  10631*  10632*  10732*  10733*  10734   10764   10765   10766   10767   10768   10769   10787*
                       10827   10841*  10842*  10843*  10844*  10845*  10846*  10866   10867   10868   10869   10870   10871
                       10887*  10897   10904*  10911   10920*  10926*  11014   11016*  11017*  11018*  11019*  11020*  11021*
                       11024*  11025*  11026   11090   11091   11092   11093   11094   11095   11147*  11243   11267*  11282*
                       11285*  11286   11299*  11302*  11303   11312   11335*  11336*  11337*  11338*  11377   11378   11379
                       11380   11405*  11406*  11407*  11408*  11409*  11416*  11417*  11418   11419   11447   11448   11449
                       11450   11451   11500   11502*  11503*  11504*  11505*  11506*  11507*  11510*  11511*  11512   11597
                       11598   11599   11600   11601   11602   11634*  11744   11788*  11789*  11790*  11791*  11792*  11793*
                       11858   11859   11860   11861   11862   11863   11878*  11898   11948*  11951   11953   11954   11982
                       11983   11987*  12003*  12004*  12005*  12006*  12007*  12008*  12009   12012*  12025   12027*  12029
                       12039   12041   12043   12044   12045   12046   12047   12049*  12050*  12078*  12079   12080*  12082
```

```
                    12083   12084*  12086   12088   12090   12096   12098*  12100*  12108*  12112   12116   12117   12121
                    12162*  12163*  12164   12168   12175   12177   12180   12183*  12184*  12185   12190   12192   12194*
                    12195   12213*  12214*  12215*  12216*  12217*  12223*  12236*  12241   12252   12253*  12254*  12255*
                    12285*  12286*  12287*  12288*  12289*  12291   12295*  12297   12299   12307*  12308   12310   12311*
                    12313   12314   12315   12317   12351   12364*  12367*  12379*  12384*  12403*  12409*  12413*  12420
                    12421   12452*  12453   12454   12455*  12460*  12461*  12462*  12468   12493   12494   12495   12496*
                    12497*  12512*  12513   12550*  12551*  12552*  12553*  12554*  12555*  12556*  12557   12565*  12569
                    12570   12571   12572   12573   12574   12575   12580*
SR0     = 177572    1918#
SR1     = 177574    1919#
SR2     = 177576    1920#
SR3     = 172516    1921#
SSECTR  047602      11623#  11630*
SSYN    044424      11030*  11040#  11175*
STACK   = 001000    1785#   2839    2934    2981    3213    3239    3265    3292    3318    3406    3432    3458    3484
                    3510    3543    3601    3951    4012    4262    4348    4381    4439    4511    4615    4839    5122
                    5363    5491    5636    5772    5907    5976    5999    6204    6227    6331    6447    6566    6658
                    6780    6901    7024    7113    7237    7373    7504    7621    7736    7837    7980    8119    8223
                    8330    8416    8443    8500    8528    8583    8613    8667    8694    8757    8825    8893    8958
                    9014    9038    9144    9275    9300    9408    9521
START   004222      2828    2632#
STKLMT= 177774      1796#
SWR     001140      1978#   2837    2858*   2860    2866*   2951    3021    3042    4122    5390    7091    7207    7711
                    8186    9103    9258    9819    9887    10138   10725   10727   11943   11957   11959   11965   11972
                    12347   12354   12359   12362   12556   12569*
SWREG   000176      1901#   2866
SW0     = 000001    1849#
SW00    = 000001    1839#   1849
SW01    = 000002    1838#   1848
SW02    = 000004    1837#   1847
SW03    = 000010    1836#   1846
SW04    = 000020    1835#   1845
SW05    = 000040    1834#   1844
SW06    = 000100    1833#   1843
SW07    = 000200    1832#   1842    4123    4124    7093    7209    7713    8188    9105    9260    10140
SW08    = 000400    1831#   1841    4123
SW09    = 001000    1830#   1840    9819    9887
SW1     = 000002    1848#
SW10    = 002000    1829#
SW11    = 004000    1828#
SW12    = 010000    1827#
SW13    = 020000    1826#   2951    3021    3042    5390
SW14    = 040000    1825#
SW15    = 100000    1824#
SW2     = 000004    1847#
SW3     = 000010    1846#
SW4     = 000020    1845#
SW5     = 000040    1844#
SW6     = 000100    1843#   10727
SW7     = 000200    1842#
SW8     = 000400    1841#   10725
SW9     = 001000    1840#
TAGDTE  002014      2785#
TBITVE= 000014      1882#
TDF     = 000040    2611#
TESDTE  002012      2780#   11187   11687   11941*
```

```
TESTAD  036706          9722#
TIMCNT  040420         10004#  10013   10018
TKVEC =  000060          1889#  12149*  12150*
TMPILL  002006          2773#   8231*   8234    8236*   8240    8242    8243*
TOLGAP  047336          7031    8129    9045    9151   11411   11486#
TOTALA  002004          2770#   3036*   3038*   4864    5012
TPVEC =  000064          1890#
TRAPVE=  000034          1888#   2845*   2846*
TRE     =  040000          2518#   3969    3978    7784    7786    8295    9199    9353
TRK     044364         11026*  11028#
TRK1    =  004000          2598#
TRK10 =  040000          2601#
TRK2    =  010000          2599#
TRK20 =  100000          2602#
TRK4    =  020000          2600#
TRTVEC=  000014          1883#
TSECC   002010          2776#  11420   11879   11889   11933*
TSECCG  042526         10597#  10718   11275   11287   11292   11304   11353   11368   11422*  11434*  11881*  11891*  11937*
TSTNM   004172          2815#   2936*   2984*   3000*   3176*   3216*   3242*   3268*   3295*   3321*   3347*   3409*   3435*
                        3461*   3487*   3513*   3545*   3604*   3953*   4014*   4073*   4140*   4178*   4260*   4298*   4346*
                        4383*   4441*   4513*   4617*   4698*   4842*   5124*   5365*   5493*   5638*   5774*   5909*   6001*
                        6230*   6334*   6450*   6569*   6661*   6783*   6904*   7027*   7116*   7240*   7376*   7507*   7624*
                        7739*   7840*   7983*   8122*   8226*   8333*   8446*   8531*   8616*   8697*   8760*   8828*   8896*
                        8961*   9041*   9146*   9302*   9410*   9523*   9729   13224   13226   13228   13230   13233   13239
                       13242   13245   13247   13255   13261   13264   13268   13271   13274   13277   13280
TST1    005414          2915    2932#   9694
TST10   007764          3236#
TST100  032662          8499#
TST101  032710          8525#
TST102  033032          8582#
TST103  033060          8610#
TST104  033160          8666#
TST105  033206          8691#
TST106  033306          8754#
TST107  033434          8822#
TST11   010030          3262#
TST110  033562          8890#
TST111  033666          8955#
TST112  033772          9013#
TST113  034020          9036#
TST114  034422          9143#
TST115  035144          9274#
TST116  035172          9299#
TST117  035516          9407#
TST12   010074          3289#
TST120  036034          9520#
TST121  036340          9627#
TST13   010140          3315#
TST14   010204          3344#
TST15   010372          3403#
TST16   010436          3429#
TST17   010502          3455#
TST2    005670          2979#
TST20   010546          3481#
TST21   010612          3507#
TST22   010656          3542#
```

```
TST23   011026          3600#
TST24   012264          3926    3950#
TST25   012434          3955    3992    4011#
TST26   012654          4016    4054    4071#
TST27   013106          4075    4138#
TST3    005742          2997#
TST30   013224          4142    4160    4175#
TST31   013516          4180    4237    4255#
TST32   013636          4280    4296#
TST33   014002          4328    4344#
TST34   014126          4366    4380#
TST35   014252          4402    4438#
TST36   014414          4510#
TST37   014772          4614#
TST4    006014          3007    3017#
TST40   015216          4696#
TST41   015606          4838#
TST42   016732          5093    5121#
TST43   017606          5362#
TST44   020246          5490#
TST45   020654          5635#
TST46   021220          5771#
TST47   021564          5906#
TST5    007120          3044    3120    3131#   9651    9660
TST50   021742          5975#
TST51   021770          5998#
TST52   022650          6203#
TST53   022676          6226#
TST54   023160          6285    6308    6330#
TST55   023470          6392    6419    6446#
TST56   024002          6510    6537    6565#
TST57   024240          6657#
TST6    007356          3174#
TST60   024576          6730    6755    6779#
TST61   025134          6852    6877    6900#
TST62   025472          6973    6998    7022#
TST63   026016          7074    7095    7112#
TST64   026354          7168    7211    7230#
TST65   026704          7342    7365#
TST66   027212          7472    7502#
TST67   027570          7599    7620#
TST7    007720          3181    3210#
TST70   030126          7715    7735#
TST71   030456          7741    7829#
TST72   031026          7951    7977#
TST73   031402          8052    8093    8115#
TST74   031724          8197    8220#
TST75   032220          8327#
TST76   032512          8415#
TST77   032540          8440#
TTNO  = 000120          2936#   2984#   3000#   3176#   3216#   3242#   3268#   3295#   3321#   3347#   3409#   3435#   3461#
                        3487#   3513#   3545#   3604#   3953#   4014#   4073#   4140#   4178#   4260#   4298#   4346#   4383#
                        4441#   4513#   4617#   4698#   4842#   5124#   5365#   5493#   5638#   5774#   5909#   6001#   6230#
                        6334#   6450#   6569#   6661#   6783#   6904#   7027#   7116#   7240#   7376#   7507#   7624#   7739#
                        7840#   7983#   8122#   8226#   8333#   8446#   8531#   8616#   8697#   8760#   8828#   8896#   8961#
                        9041#   9146#   9302#   9410#   9523#
```

```
TUF    = 000100          2612#
TY       044530         11136#
TYPDS  = 104404          3094    3109    5397    9635    9641    9649    9683   12414   12532#
TYPE   = 104400          2880    2884    2888    2892    2896    2900    2904    2908    2917    2953    2957    3023    3046
                         3050    3054    3058    3063    3089    3095    3101    3107    3138    3144    3145    3151    3152
                         3158    3183    3187    3191    3197    5392    9630    9636    9647    9650    9681    9684    9725
                         9731    9737    9738    9742    9746    9753    9757   10883   10889   10900   10906   10912   10916
                        10922   10928   11905   12048   12091   12166   12174   12244   12247   12251   12319   12321   12349
                        12357   12378   12395   12397   12400   12402   12417   12422   12487   12528#  12578
TYPOC  = 104401          2963    3100    3143    3150    3157    9730    9736   10888   10905   10921   10927   11909   12386
                        12410   12529#
TYPON  = 104403         12531#
TYPOS  = 104402         12530#
T.SCOP   040046          9887#  12538
UN       005732          2986*   2987*   2989#
UNIT     001760          2742#   2923*   3069*   3114*   3122*   3136    3142    3220    3246    3272    3299    3325    3413
                         3439    3465    3491    3517    3567    3710    3726    3756    3958    3988    4018    4033    4088
                         4182    4202    4239    4305    4311    4316    4321    4541    4545    4767    4874    4923    5201
                         5309    5396    5434    5579    5714    5851    7290    7424    7529    7791    7892    8339    9634
                         9654    9659*   9824    9909
UNITS    001740          2741#   3067*   3072    3076    3114    3648    9655
UNITSL   001770          2748#   2924*   3122
UNLOAD   002020          2794#   5373
UNS    = 040000          2555#
UPE    = 020000          2494#   4318
US1    = 000001          2481#
US2    = 000002          2482#
US4    = 000004          2483#
UWR    = 000010          2646#
VUF    = 000002          2645#
VU30   = 010000          2618#
VV     = 000100          2529#   3860    4460    4647    7540    7559    7588    8268    9111    9114    9943
WAIT.T   040422         10005#  12540
WAT    = 104412          4030    4198    4713    4716    4781    4784    8290    9122   12540#
WC       001672          2717#   4705    4739    4773    4807    7908    9782   13239
WCE    = 040000          2495#   7295    7429    7897
WCF    = 000040          2546#
WCRC     046312          6695    6817    6938    7044    7139    7647    7759    8017    8767*   8835*   8905*   8970*   9058
                         9164   10080   10169   10323   11185   11478#
WCU    = 000001          2606#
WCYL     047516          6242*   6249    6345*   6352    6462*   6469    6581*   6588    8140   11523#
WECC1    047330          6288*   6395*   6513*  11483#  13271
WECC2    047332          6289*   6396*   6514*  11484#  13271
WKEY1    047522          6245*   6348*   6465*   6584*  11525#
WKEY2    047524          6246*   6349*   6466*   6585*  11526#
WLE    = 004000          2552#
WORD     045322         11165*  11169*  11177*  11179*  11181*  11183*  11185*  11226*  11230*  11260#  11270*  11283*  11300*
                        11885*  11893*
WRCHDA   041100          7417    7885   10159#
WRCHDT   002034          2800#  10096
WRCHEK   002032          2799#  10186
WRCHHD   040574          7285   10071#
WRDATA   046000         11082   11401#  11593
WRFROM   002062          2813#   4467    4473*   4480    4568    4574*   4580*   4586    4654    4660*   4667    4727    4733*
                         4740    4795    4801*   4808    5167    5175*   5182    5219    5226*   5232    5279    5286*   5293
                         5326    5332*   5338    5404    5411*   5418    5449    5456*   5461    5548    5555*   5562    5595
```

```
                    5601*  5607   5684   5690*  5697   5729   5735*  5741   5821   5827*  5834   5866   5872*
                    5878   5941   5947*  5954   6175   6181*  6183   6255   6358   6475   6594   6733   6749
                    6855   6871   6976   6992   7051   7052   7056   7171   7172   7176   7177   7182   7309
                    7316   7317   7321   7328   7329   7336   7445   7451   7458   7459   7465   7677   7678
                    7682*  7686   7924   7930   7937   7938   7944   8055   8071   8146   8147   8151   8346
                    8380   8386*  8393   8462   8471   8472   8547   8556   8557   8629   8630   8653   8710
                    8711   8734   8778   8787   8788   8846   8855   8856   8911   8912   8935   8976   8977
                    9000   9065   9066   9070   9169   9170   9174   9195   9212   9218*  9225   9363   9369*
                    9376   9423   9475   9481*  9488   9536   9583   9589*  9596   10386  10486
WRHEAD  047612       11522  11629#
WRIDAT  002036        2801#  7063   8156   9077   9181   9550
WRIFOR  002040        2802#  6269   6377   6494   6608   8343   9442   10483
WRITE   045560       11194  11205  11334#  11424  11430  11436  11440  11443  11673  11689  11702  11714  11725
                     11735
WRL   = 004000        2534#
WRU   = 000400        2614#
WSECTR  047520        6244*  6347*  6464*  6583*  11510  11524#
WSSYNC  046300       11476#
WSU   = 000004        2608#
WTRK    047454       11512*  11514#
WWORD   045556       11195  11199  11202  11206  11213  11218  11222  11329#  11376*  11425  11431  11437  11441
                     11444  11674  11678  11691  11703  11715  11726  11736
X       044422        6692*  6814*  6935*  7041*  7136*  7644*  7756*  8014*  8138*  9055*  9161*  10077*  10165*
                     10360*  10376  11038#  11075
XE2     006634        3041   3070#
Y       044462       11084#
ZCODE   042542       10609#  10799*  13280
ZER   = 000400        2567#  10808
ZWORDS  045776       11400#  11418*
$AUTOB  001134        1975#
$BDADR  001122        1970#   3963*  3972*  3984*  3994*  4023*  4038*  4093*  4154*  4187*  4207*  6621*  8251*
                      8261*   8275*  8298*  9928*  9934*  9938*  9948*  9953*  9956*  9973*  9980*  9986*  9991*
                     13230  13233  13252
$BDDAT  001126        1972#   3004*  3005   3360*  3382*  3557*  3558   3570*  3571   3577*  3578   3717*  3718
                      3733*  3734   3744*  3745   3758*  3759   3770*  3771   3780*  3781   3790*  3791   3800*
                      3801   3810*  3811   3820*  3821   3830*  3831   3838*  3840   3849*  3850   3859*  3871*
                      3872   3883*  3884   3896*  3897   3907*  3908   4048*  4057*  4106*  4156*  4158   4222*
                      4230*  4242*  4268*  4269   4277*  4278   4307*  4308   4324*  4325   4355*  4356   4363*
                      4364   4391*  4392   4399*  4400   4890*  4891   4901*  4902   4912*  4913   4925*  4926
                      4943*  4945   4954*  4955   4964*  4965   4974*  4975   4984*  4985   4994*  4995   5004*
                      5005   5014*  5015   5023*  5024   5033*  5043*  5044   5054*  5055   5063*  5064   5074*
                      5075   6050*  6052   6120*  6123   7083*  7189*  7555*  7693*  7782*  7784   7794*  7795
                      7800*  7802   8174*  9095*  9115*  9250*  9832*  9834*  9836   9838*  10023*  10130*  10429*
                     10430  10439*  10440  10451*  10452  10459*  10461  10530*  10531  11199*  11218*  11668*  11678*
                     11696*  11709*  11718*  11730*  11740*  13224  13228  13233  13237  13247  13249  13255  13258
                     13261  13264
$BELL   001216        2004#  12174  12260  12349  12370
$CHARC  052004       12093*  12103*  12110  12119*  12124#
$CKSWR= ****** U     12535
$CMTAG  001100        1958#   2834   2835   2843   2849   2850
$CM1  = 000006        1990#   1991#  1992#  1993#  1994#  1995#  1996#
$CM2  = 000014        1990#   1991#  1992#  1993#  1994#  1995#  1996#
$CM3  = 000006        1988#   1990
$CM4  = 000006        1996#   1997#  1998#  1999#  2000#  2001#  2002#
$CNTLC  052507       12166  12260#
$CNTLG  052521       12262#
```

DERPSB.P11 CROSS REFERENCE TABLE -- USER SYMBOLS

```
$CNTLU  052514          12261#
$CRLF   001223           2006#  3107   3144   3151   3158   9737   12092  12127  12260  12324  12357  12370  12378
                        12397  12402  12422
$DBLK   051560          12014  12048  12056#
$DOAGN  036662           9677   9686   9692#
$DTBL   051550          12017  12052#
$ENDAD  036652           1947   9688#
$ENDCT  036620           9679#
$ENDMG  036671           9647   9681   9696#
$ENULL  036666           9650   9684   9695#
$EOP    036564           2964   3062   9653   9669#
$ECPCT  036612           9676#  9680
$ERFLG  001103           1961#  9821   11922  11961  11963  11969* 11990  12344* 12370
$ERMAX  001115           1967#  2851*  11963  11985* 11990
$ERROR  052710           2843   2965   12338#
$ERRPC  001116           1968#  12351* 12352* 12353  12370  12384  13224  13226  13228  13230  13233  13236  13237
                        13239  13242  13245  13247  13249  13252  13255  13258  13261  13264  13268  13271  13274
                        13277  13280
$ERRTB  001226           2023#  12392
$ERRTY  053052          12356  12377#
$ERTTL  001112           1965#  9640   9642*  12350* 12370
$ESCAP  001214           2003#  2850*  11984* 12365  12367  12370
$FILLC  001156           1986#  12096  12127
$FILLS  001155           1985#  12127
$GDADR  001120           1969#
$GDDAT  001124           1971#  3009*  3357*  3381*  3561*  3569*  3571   3581*  3715*  3718   3732*  3743*  3757*
                        3759   3769*  3779*  3789*  3799*  3809*  3819*  3829*  3839*  3840   3848*  3857*  3870*
                        3882*  3895*  3906*  4046*  4056*  4107*  4157*  4158   4221*  4229*  4241*  4271*  4282*
                        4311*  4312*  4323*  4358*  4368*  4394*  4404*  4889*  4900*  4911*  4924*  4926   4940*
                        4942*  4945   4953*  4963*  4973*  4983*  4993*  5003*  5012*  5013*  5015   5022*  5031*
                        5042*  5053*  5062*  5073*  7082*  7188*  7554*  7692*  7786*  7793*  7795   7804*  8173*
                        9094*  9114*  9249*  9827*  9829*  9831   9836   9841*  9847*  10129* 10434* 10445* 10455*
                        10464* 10533* 11174* 11198* 11215* 11217* 11240* 11667* 11677* 11695* 11708* 11719* 11729*
                        11741* 13224  13226  13228  13247  13255  13258  13261  13264  13268
$GET42  036642           9685#
$GTSWR= ******  U       12534
$HD   = 000000           1768
$HIOCT  052706          12312* 12323#
$ICNT   001104           1962#  11976* 11977  11979* 11989
$ILLUP  053670          12548  12564  12583#
$INTAG  001135           1976#
$ITEMB  001114           1966#  12353* 12370  12381
$LF     001224           2007#  12127  12251  12260  12324  12370
$LPADR  001106           1963#  2852*  3133*  9752*  9763   11967* 11982* 11987  11989
$LPERR  001110           1964#  2853*  3690*  4757*  7522*  7577*  8247*  9735   9762*  11967  11983*  11989  12364
$MAIL = ******  U        2870   11982  12080  12359
$MNEW   052537          12265#
$MSWR   052526          12263#
$MXCNT  051342          11980  11989#
$NULL   001154           1984#  12098  12127
$NWTST= 000001           2928#  2930   2972#  2974   2994#  3014#  3124#  3126   3166#  3168   3204#  3206   3230#
                        3232   3256#  3258   3283#  3285   3309#  3311   3338#  3340   3397#  3399   3423#  3425
                        3449#  3451   3475#  3477   3501#  3503   3537#  3539   3593#  3595   3940#  3942   3997#
                        3999   4063#  4065   4132#  4134   4167#  4169   4249#  4251   4290#  4292   4337#  4339
                        4374#  4376   4429#  4431   4502#  4504   4605#  4607   4688#  4690   4828#  4830   5112#
                        5114   5354#  5356   5481#  5483   5626#  5628   5762#  5764   5897#  5899   5972#  5987#
```

```
                    5989    6200#   6216#   6218    6320#   6322    6437#   6439    6556#   6558    6649#   6651    6771#
                    6773    6892#   6894    7013#   7015    7103#   7105    7221#   7223    7355#   7357    7489#   7491
                    7610#   7612    7725#   7727    7817#   7819    7968#   7970    8106#   8108    8213#   8215    8309#
                    8311    8412#   8427#   8429    8496#   8512#   8514    8579#   8597#   8539    8663#   8678#   8680
                    8741#   8743    8809#   8811    8877#   8879    8942#   8944    9010#   9027#   9029    9129#   9131
                    9271#   9288#   9290    9394#   9396    9507#   9509    9618#   9620
$OCNT   053450      12459*  12488*  12501#
$OMODE  053452      12454*  12458*  12463   12466*  12477*  12503#
$OVER   051326      11944   11960   11968   11978   11986#
$PASS   001100       1959#   5388    9646*   9648    9673*   9674*   9682    9695    11974   11990
$POWER  053676      12579   12586#
$PWRAD  053664      12581#
$PWRDN  053524       2847   12548#  12576
$PWRMG  053660      12579#
$PWRUP  053576      12558   12564#
$QUES   001222       2005#  12127   12244   12260   12321   12324   12370
$RDCHR  052300      12213#  12535
$RDDEC= ****** U    12538
$RDLIN  052370      12236#  12536
$RDOCT  052550      12285#  12537
$RDSZ = 000011      12229#
$REGAD  001160       1988#
$REG0   001162       1990#
$REG1   001164       1991#
$REG2   001166       1992#
$REG3   001170       1993#
$REG4   001172       1994#
$REG5   001174       1995#
$RTNAD  036664       9694#
$R2A  = ****** U    12538
$SAVRE= ****** U    12538
$SAVR6  053674      12557*  12565   12566*  12567*  12585#
$SCOPE  051054       2841   11931#
$SETUP= 000017       2833#   2840    2841    2843    2845    2847    2849    2850    2852    9671    11932   12170   12171
                    12202   12267   12339   12362   12369
$SS1  = 000000       2870#
$STUP = 177777       2833#
$SVLAD  051300      11952   11981#
$SWR  = 167700       1758#   1768    1773    1774    1775    1776    1777    1778    1779    2002    2003    2004    2849
                     2850    2852    2853    2933    2980    2998    3018    3132    3175    3211    3237    3263    3290
                     3316    3345    3404    3430    3456    3482    3508    3543    3601    3951    4012    4072    4139
                     4176    4256    4297    4345    4381    4439    4511    4615    4697    4839    5122    5363    5491
                     5636    5772    5907    5976    5999    6204    6227    6331    6447    6566    6658    6780    6901
                     7023    7113    7231    7366    7503    7621    7736    7830    7978    8116    8221    8328    8416
                     8441    8500    9526    8583    8611    8667    8692    8755    8823    8891    8956    9014    9037
                     9144    9275    9300    9408    9521    9628    9666    9672    9687    9693    9695    11923   11924
                    11925   11926   11927   11943   11955   11957   11958   11961   11962   11963   11970   11971   11972
                    11983   11986   11989   12330   12331   12332   12333   12334   12347   12354   12359   12362   12370
                    12582
$SWRMK= 000000       1779    1780    11927   11928   11959
$TIMES  001212       2002#   2849*   2933*   2980*   2998*   3018*   3132*   9628*   9672*   11970*  11977   11980*  11989
$TKB    001146       1981#   12130   12151   12162   12183
$TKCNT  052010      12131#  12146*  12172   12189*  12220   12222*
$TKINT  052030       2874    3020    4540    9825    12146#  12167
$TKQEN= 052027      12135#  12197   12225
$TKQIN  052012      12132#  12147*  12148   12195*  12196*  12197   12199*
```

```
$TKQOU  052014          12133#  12148*  12223   12224*  12225   12227*
$TKQSR  052016          12134#  12147   12199   12227
$TKS    001144           1980#  12130   12152*  12179*  12181   12187*
$TKSRV  052100          12149   12162#
$TMP0   001176           1996#   3080*   3081*  10009*  10014   10019   10023   10120*  10134   10227*  10273*  13237
$TMP1   001200           1997#   2949*   6027*   6052    6119*   6123    6137*   6167    7524*   7570*  10010*  10014   10019
                        10121*  10136   13236   13237   13249
$TMP2   001202           1998#   6055*   6056*   6125*   6126*  10013*  10016*  10018*  10021*  10229*  10271*  13249
$TMP3   001204           1999#   6059*   6129*  10007*  10008*  10228*  10230   10231*  13237   13249
$TMP4   001206           2000#   6049*   6058    6080*   6109*   6128   10230*  10254*
$TMP5   001210           2001#   5140*   5142*   5143    5211   10226*  10232*  10238   10349*  10427   10449
$TN   = 000122           1758#   1768    2914    2928    2933#   2935    2972    2980#   2983    2994    2998#   2999    3006
                         3014    3018#   3044    3119    3124    3132#   3166    3175#   3180    3204    3211#   3215    3230
                         3237#   3241    3256    3263#   3267    3283    3290#   3294    3309    3316#   3320    3338    3345#
                         3346    3397    3404#   3408    3423    3430#   3434    3449    3456#   3460    3475    3482#   3486
                         3501    3508#   3512    3537    3543#   3544    3593    3601#   3603    3924    3940    3951#   3952
                         3955    3991    3997    4012#   4013    4016    4053    4063    4072#   4075    4132    4139#   4142
                         4159    4167    4176#   4177    4180    4236    4249    4256#   4259    4279    4290    4297#   4327
                         4337    4345#   4365    4374    4381#   4382    4401    4429    4439#   4440    4502    4511#   4512
                         4605    4615#   4616    4688    4697#   4828    4839#   4841    5091    5112    5122#   5123    5354
                         5363#   5364    5481    5491#   5492    5626    5636#   5637    5762    5772#   5773    5897    5907#
                         5908    5972    5976#   5987    5999#   6000    6200    6204#   6216    6227#   6229    6284    6307
                         6320    6331#   6333    6391    6418    6437    6447#   6449    6509    6536    6556    6566#   6568
                         6649    6658#   6660    6729    6754    6771    6780#   6782    6851    6876    6892    6901#   6903
                         6972    6997    7013    7023#   7026    7073    7094    7103    7113#   7115    7167    7210    7221
                         7231#   7239    7341    7355    7366#   7375    7471    7489    7503#   7506    7598    7610    7621#
                         7623    7714    7725    7736#   7738    7741    7817    7830#   7839    7950    7968    7978#   7982
                         8051    8092    8106    8116#   8121    8196    8213    8221#   8225    8309    8328#   8332    8412
                         8416#   8427    8441#   8445    8496    8500#   8512    8526#   8530    8579    8583#   8597    8611#
                         8615    8663    8667#   8678    8692#   8696    8741    8755#   8759    8809    8823#   8827    8877
                         8891#   8895    8942    8956#   8960    9010    9014#   9027    9037#   9040    9129    9144#   9145
                         9271    9275#   9288    9300#   9301    9394    9408#   9409    9507    9521#   9522    9618    9628#
$TPB    001152           1983#  12116*  12127
$TPFLG  001157           1987#  12074   12127
$TPS    001150           1982#  12114   12127
$TRAP   053454           2845   12512#
$TRP  = 000013          12520#  12529#  12530#  12531#  12532#  12533#  12535   12536#  12537#  12538#  12539#  12540#  12541#
$TRPAD  053476          12517   12527#
$TSTNM  001102           1960#   9645*   9671*  11922   11959   11981*  11986   11990   12346   12370
$TTYIN  052476          12237   12238   12255   12259#
$TYPBN= ****** U        12533
$TYPDS  051344          12002#  12532
$TYPE   051570          12074#  12520   12528
$TYPEC  051740          12095   12102   12109   12114#  12115
$TYPEX  052006          12120   12122   12125#
$TYPOC  053252          12457#  12529
$TYPON  053266          12456   12459#  12531
$TYPOS  053226          12452#  12530
$XTSTR  051104          11946#
$$GET4= 000000           9687#
$OFILL  053451          12453*  12457*  12467   12502#
$4DCAT= ****** U        11943   12356
.     = 063532           1895#   1899#   1904#   1946#   1948#   1950#   1957#   2008    2741#   2813#   2814#   2838    2852
                         2853    2891#   2895#   2899#   2956#   3049#   3053#   3066#   3092#   3141#   3186#   3190#   3200#
                         5395#   9639#   9695    9699    9741#   9756#  10886#  10892#  10903#  10909#  10919#  11475#  11476#
                        11477#  11478#  11479#  11480#  11482#  11483#  11484#  11485#  11486#  11989   11990   12056#  12127
```

12130 12134# 12135 12136# 12259# 12260 12267 12324 12370 12425# 12560 12584 13223#
13338#.

```
CHECKB   1769#  5104   6019   6265   6372   6490   6604   9329   9437   9545  10345
COMMEN      1#  1892#  3659
ENDCOM      1#  1892#  3674
ERROR    1786#  2950   2991   3011   3224   3250   3276   3303   3329   3362   3384   3417   3443   3469   3495
         3521   3563   3574   3583   3654   3656   3720   3931   3964   3973   3985   3995   4024   4039   4050
         4059   4094   4111   4113   4155   4163   4188   4208   4223   4231   4244   4273   4284   4314   4331
         4360   4370   4396   4406   4489   4595   4676   4749   4817   5097   5192   5242   5301   5348   5427
         5471   5571   5617   5706   5751   5843   5884   5963   6060   6130   6197   6310   6312   6421   6423
         6539   6541   6622   6757   6759   6879   6981   7000   7002   7087   7089   7193   7195   7297   7303
         7344   7346   7431   7436   7475   7477   7535   7543   7556   7562   7583   7593   7601   7697   7699
         7788   7798   7806   7899   7904   7911   7917   7954   7956   8077   8079   8095   8178   8184   8199
         8252   8262   8276   8299   8401   9099   9101   9116   9234   9254   9256   9385   9497   9605   9929
         9935   9939   9949   9954   9957   9974   9981   9987   9992  10024  10392  10394  10418  10420  10435
        10446  10456  10466  10535  10738  10744  10812
ESCAPE      1#  1892#
GETPRI      1#  1892#
GETSWR      1#  1892#
HCOMPR   1769#  8442   8527   8756   8824
HCOMPW   1769#  8612   8693   8892   8957
MAKECL   1769#  5972   6200   8412   8496   8579   8663   9010   9271
MSG      2928#  2930   2971#  2974   3124#  3126   3166#  3168   3204#  3206   3230#  3232   3256#  3258   3283#
         3285   3309#  3311   3337#  3340   3397#  3399   3423#  3425   3449#  3451   3475#  3477   3501#  3503
         3535#  3539   3592#  3595   3939#  3942   3997#  3999   4062#  4065   4131#  4134   4166#  4169   4247#
         4251   4289#  4292   4335#  4339   4373#  4376   4428#  4431   4501#  4504   4604#  4607   4687#  4690
         4827#  4830   5111#  5114   5353#  5356   5480#  5483   5625#  5628   5761#  5764   5896#  5899   5986#
         5989   6215#  6218   6319#  6322   6436#  6439   6555#  6558   6648#  6651   6770#  6773   6891#  6894
         7012#  7015   7102#  7105   7218#  7223   7354#  7357   7487#  7491   7609#  7612   7724#  7727   7816#
         7819   7967#  7970   8105#  8108   8212#  8215   8308#  8311   9026#  9029   9128#  9131   9287#  9290
         9393#  9396   9506#  9509   9618#  9620
MSGA     8427#  8429   8514   8743   8811
MSGB     8596#  8599   8680   8879   8944
MULT        1#  1892#
NEWTST      1#  1892#  2928   2972   2994   3014   3124   3166   3204   3230   3256   3283   3309   3338   3397
         3423   3449   3475   3501   3537   3593   3940   3997   4063   4132   4167   4249   4290   4337   4374
         4429   4502   4605   4688   4828   5112   5354   5481   5626   5762   5897   5972   5987   6200   6216
         6320   6437   6556   6649   6771   6892   7013   7103   7221   7355   7489   7610   7725   7817   7968
         8106   8213   8309   8412   8427   8496   8512   8579   8597   8663   8678   8741   8809   8877   8942
         9010   9027   9129   9271   9288   9394   9507   9618
POP         1#  1892#  9787   9880  10058  10144  10276  10324  10499  10537  10763  10826  10866  11089  11242
        11312  11377  11447  11596  11743  11857  11897  12043  12313  12569  12570
PUSH        1#  1892#  9777   9867  10048  10111  10218  10221  10296  10480  10515  10626  10787  10840  11016
        11147  11266  11334  11405  11502  11634  11788  11878  12002  12287  12550  12556
REPORT      1#  1892#
RFORGC   1769#  6307   6418   6536   6754   6876   6997   7341   7471   7950
SAVE     1769#  12339
SAVTST   1769#  2935   2983   2999   3175   3215   3241   3267   3294   3320   3346   3408   3434   3460   3486
         3512   3544   3603   3952   4013   4072   4139   4177   4259   4297   4345   4382   4440   4512   4616
         4697   4841   5123   5364   5492   5637   5773   5908   6000   6229   6333   6449   6568   6660   6782
         6903   7026   7115   7239   7375   7506   7623   7738   7839   7982   8121   8225   8332   8445   8530
         8615   8696   8759   8827   8895   8960   9040   9145   9301   9409   9522
SCOPE    1787#  2932   2979   2997   3017   3131   3174   3210   3236   3262   3289   3315   3344   3403   3429
         3455   3481   3507   3542   3600   3950   4011   4071   4138   4175   4255   4296   4344   4380   4438
         4510   4614   4696   4838   5121   5362   5490   5635   5771   5906   5975   5998   6203   6226   6330
         6446   6565   6657   6779   6900   7022   7112   7230   7365   7502   7620   7735   7829   7977   8115
         8220   8327   8415   8440   8499   8525   8582   8610   8666   8691   8754   8822   8890   8955   9013
         9036   9143   9274   9299   9407   9520   9627   9670
```

```
SETPRI        1#   1892#  12216
SETTRA    12520#  12529   12530  12531  12532  12535  12536  12537  12538  12539  12540
SETUP         1#   1892#   2833
SKIP          1#   1769#   1892#  2914   3006   3119   3180   3924   3991   4053   4159   4236   4279   4327   4365
              4401  5091   6284   6391   6509   6729   6851   6972   7073   7094   7167   7210   7598   7714   8051
              8092  8196
SLASH         1#   1892#
SMORE      1769#  11932
SPACE      1892#
STARS         1#   1892#  1945   1953   2008   2463   2468   2502   2504   2928   2931   2972   2978   2994   2996
              3014  3016   3124   3130   3166   3173   3204   3209   3230   3235   3256   3261   3283   3288   3309
              3314  3338   3343   3397   3402   3423   3428   3449   3454   3475   3480   3501   3506   3537   3541
              3593  3599   3940   3949   3997   4010   4063   4070   4132   4137   4167   4174   4249   4254   4290
              4295  4337   4343   4374   4379   4411   4421   4429   4437   4502   4509   4605   4613   4688   4695
              4828  4837   5112   5120   5354   5361   5481   5489   5626   5634   5762   5770   5897   5905   5972
              5974  5987   5997   6200   6202   6216   6225   6320   6329   6437   6445   6556   6564   6627   6644
              6649  6656   6771   6778   6892   6899   7013   7021   7103   7111   7221   7229   7355   7364   7489
              7501  7610   7619   7725   7734   7817   7828   7968   7976   8106   8114   8213   8219   8309   8326
              8412  8414   8427   8439   8496   8498   8512   8524   8579   8581   8597   8609   8663   8665   8678
              8690  8741   8753   8809   8821   8877   8889   8942   8954   9010   9012   9027   9035   9129   9142
              9271  9273   9288   9298   9394   9406   9507   9519   9617   9618   9626   9663  10951  10952  11097
             11119 11126  11128  11253  11255  11320  11322  11388  11390  11461  11467  11493  11495  11610  11612
             11754 11756  11867  11869  11919  11992  12059  12129  12205  12229  12273  12326  12372  12429  12506
             12546 12562
SWRSU         1#   1892#  2854#
TJUMP      1769#   3044   3955   4016   4075   4142   4180   7741
TRMTRP    12520#
TSCLR      1769#   3731   3742   3768   3778   3788   3798   3808   3818   3828   3847   3894   3905
TSCLR1     1769#   3869   3881
TSCLR2     1769#   4888   4899   4910   4952   4962   4972   4982   4992   5002   5021   5061   5072
TSCLR3     1769#   5041   5052
TSCLR4     1769#   3754
TSCLR5     1769#   4922
TYPBIN        1#   1892#
TYPDEC        1#   1892#  9682   12412
TYPNAM        1#   1892#
TYPNUM        1#   1892#
TYPOCS        1#   1892#
TYPOCT        1#   1892#  3149   3156   12384  12409
TYPTXT        1#   1892#  2879   2884   2888   2892   2896   2900   2904   2908   2917   2953   2957   3023   3045
              3050  3054   3058   3063   3089   3095   3101   3138   3145   3152   3183   3187   3191   3197   5392
              9630  9636   9725   9731   9738   9742   9746   9753   9757   10882  10889  10900  10906  10912  10916
             10922 10928  11904
WDATAR     1769#   7023   9037
XREAD      1769#   6663   6785   6906   7985
$$CMRE     1951#   1990   1991   1992   1993   1994   1995
$$CMTM     1951#   1996   1997   1998   1999   2000   2001
$$ESCA        1#   1892#
$$NEWT        1#   1892#  2928   2972   2994   3014   3124   3166   3204   3230   3256   3283   3309   3338   3397
              3423  3449   3475   3501   3537   3593   3940   3997   4063   4132   4167   4249   4290   4337   4374
              4429  4502   4605   4688   4828   5112   5354   5481   5626   5762   5897   5972   5987   6200   6216
              6320  6437   6556   6649   6771   6892   7013   7103   7221   7355   7489   7610   7725   7817   7968
              8106  8213   8309   8412   8427   8496   8512   8579   8547   8663   8678   8741   8809   8877   8942
              9010  9027   9129   9271   9288   9394   9507   9618
$$SET     12520#  12529   12530  12531  12532  12535  12536  12537  12538  12539  12540
$$SKIP        1#   1892#
```

```
.EQUAT      1#    1758#   1782
.FLOAT   1769#    3211    3237    3263    3290    3316    3404    3430    3456    3482    3508
.HEADE      1#    1758#
.KT11       1#    1758#   1910
.SETUP      1#    1758#   2832
.SWRHI      1#    1758#   1769
.SWRLO   1758#    1780#   1781
.$ACTI      1#
.$APTB      1#
.$APTH      1#
.$APTY      1#
.$ASTA      1#
.$CATC      1#    1758#   1893
.$CMTA      1#    1758#   1951
.$DB2D      1#
.$DB2O      1#
.$DIV       1#
.$EOP       1#    1758#   9661
.$ERRO      1#    1758#  12324
.$ERRT      1#    1758#  12370
.$MULT      1#
.$POWE      1#    1758#  12544
.$RAND      1#
.$RDDE      1#
.$RDOC      1#    1758#  12271
.$READ      1#    1758#  12127
.$R2AZ      1#
.$SAVE      1#
.$SB2D      1#
.$SB2O      1#
.$SCOP      1#    1758#  11917
.$SIZE      1#
.$SUPR      1#
.$TRAP      1#    1758#  12504
.$TYPB      1#
.$TYPD      1#    1758#  11990
.$TYPE      1#    1758#  12057
.$TYPO      1#    1758#  12427
.$4DCA      1#
.1170       1#
```

```
ADD     4486  4593  4674  4747  4815  5189  5240  5299  5346  5424  5469  5569  5615  5704  5749
        5841  5886  5961  6119  6133  6137  6195  8236  8243  8399  9232  9383  9495  9603  9751
        9808  9875 10238 10245 10252 12022 12084 12308 12392 12455 12465
ASL     7564 12301 12303 12305 12389 12390 12391 12516
ASLB   12027
ASR    10662 10671 10679 10687
BCC     3083  4868  7384  7403  7448  7849  7868  7927 10855 11271 11284 11301 12028
BCS     7259  7278  7325  7390  7410  7455  7565  7855  7875  7934
BEQ     2915  3007  3040  3043  3086  3088  3120  3181  3359  3380  3560  3572  3580  3623  3625
        3636  3638  3644  3719  3735  3746  3760  3772  3782  3792  3802  3812  3822  3832  3841
        3851  3862  3873  3885  3898  3909  3962  3971  3983  3992  4022  4037  4045  4054  4092
        4105  4125  4148  4160  4186  4206  4217  4219  4227  4270  4280  4310  4328  4357  4366
        4393  4402  4872  4892  4903  4914  4927  4939  4946  4956  4966  4976  4986  4996  5006
        5016  5025  5036  5045  5056  5065  5076  5139  6045  6054  6110  6124  6620  7081  7095
        7187  7211  7294  7296  7428  7430  7534  7542  7552  7561  7582  7591  7599  7691  7715
        7785  7796  7803  7896  7910  7916  8093  8172  8189  8197  8250  8260  8274  9093  9106
        9112  9248  9262  9644  9653  9657  9686  9820  9822  9837  9888  9925  9945 10127 10141
       10234 10241 10248 10426 10428 10432 10443 10450 10453 10463 10532 10636 10655 10721 10728
       10735 10741 10807 10947 11076 11190 11196 11207 11212 11214 11233 11235 11276 11278 11288
       11293 11295 11305 11344 11354 11359 11369 11421 11429 11642 11652 11665 11675 11693 11705
       11717 11727 11738 11807 11880 11958 11960 11962 11966 11975 12087 12122 12221 12296 12345
       12348 12363 12366 12394 12399 12405 12416 12482
BGE    11978
BGT     9677 12036 12193 12298 12489
BHI    11964
BHIS    4196 10804
BIC     2922  3368  3389  3627  3860  4123  4557  4645  5013  5034  5159  5207  5208  5271  5315
        5316  5386  5440  5441  5503  5541  5585  5586  5587  5649  5677  5720  5721  5786  5814
        5857  5858  6032  6037  6040  6075  6077  6104  6106  6139  6142  6153  6157  7092  7208
        7540  7559  7587  7712  8187  8267  8370  8373  9104  9110  9198  9259  9309  9417  9530
        9674  9829  9834  9846  9943 10139 10258 10263 10268 10648 10701 10707 10715 10733 10817
       10824 10896 11025 11153 11156 11159 11511 11798 11800 11826 11828 11843 11845 11853 12163
       12184 12194 12307 12479
BIS     3354  3356  3367  3378  3390  3550  3568  3696  3709  3723  3756  3959  3989  4019  4034
        4089  4183  4203  4240  4301  4306  4312  4317  4318  4322  4457  4460  4521  4541  4546
        4558  4630  4641  4647  4712  4780  4849  4875  4881  4923  5153  5157  5158  5200  5209
        5265  5269  5270  5308  5317  5382  5384  5385  5433  5442  5502  5536  5539  5540  5578
        5588  5648  5663  5675  5676  5713  5722  5785  5800  5812  5813  5850  5859  5928  5929
        5933  6035  6036  6039  6042  6074  6076  6103  6105  6138  6141  6152  6156  6162  6163
        6165  6166  6614  6616  7291  7425  7528  7663  7767  7776  7792  7893  8257  8338  8348
        8368  8369  8371  8372  9197  9199  9205  9206  9308  9350  9353  9357  9358  9416  9463
        9469  9470  9529  9571  9577  9578  9841  9847 10260 10265 10270 10488 10496 10524 10640
       10659 10668 10676 10684 10699 10700 10702 10705 10706 10708 10816 10823 10897 11023 11151
       11152 11158 11272 11274 11291 11357 11509 11639 11650 11797 11799 11804 11822 11823 11825
       11827 11842 11844 11852 12030 12031 12484 12485
BISB   12381
BIT     2951  3021  3042  3179  3195  3622  4152  4216  4235  5390  6619  7295  7429  7566  7897
        8240  8295  9819  9887  9926  9932  9946  9951  9970  9977  9984  9989 10014 10019 10233
       10240 10247 10725 10727 10808 10946 11171 11234 11237 11277 11294 11343 11358 11641 11651
       11943 11957 11965 11972 12347 12354 12362
BITB   12110
BLOS   12239
BLT    12019 12035 12101 12191 12300 12490
BMI    10257 10262 10267 12026
BNE     2838  2861  2876  2945  2952  3022  3035  3075  3113  3196  3370  3392  3617  4080  4087
        4100  4110  4128  4153  4237  4709  4770  4777  5389  5391  5519  5671  5808  6079  6083
```

```
                6108    6113    6145    6155    6236    6262    6285    6340    6369    6392    6456    6487    6510    6575    6601
                6673    6680    6730    6744    6795    6802    6852    6866    6916    6923    6973    6987    7074    7086    7098
                7168    7192    7214    7567    7571    7696    7718    7898    7995    8002    8052    8066    8166    8177    8191
                9235    8239    8241    8297    9087    9098    9108    9253    9264    9326    9434    9542    9786    9815    9840
                9844    9879    9927    9933    9947    9952    9972    9979    9985    9990   10015   10017   10020   10022   10057
               10133   10143   10272   10274   10305   10311   10375   10377   10726   10795   10797   10809   10861   10863   10899
               11074   11081   11161   11163   11168   11172   11188   11204   11229   11238   11309   11311   11373   11375   11414
               11427   11433   11439   11446   11592   11660   11662   11682   11688   11713   11734   11830   11833   11847   11849
               11851   11855   11888   11890   11896   11944   11973   12024   12081   12089   12097   12111   12118   12165   12173
               12178   12186   12198   12226   12243   12249   12355   12382   12408   12480   12568
BPL            12010   12040   12075   12115   12182   12360   12478
BR              2828    2863    2877    2881    2885    2889    2893    2897    2901    2905    2909    2918    2946    2954    2958
                3024    3047    3051    3055    3059    3064    3090    3096    3102    3105    3139    3146    3153    3184    3188
                3192    3198    3634    3639    3646    3926    4112    4149    4873    4941    5093    5141    5393    6073    6135
                7088    7194    7298    7432    7568    7698    7900    7913    8183    8237    8303    9100    9255    9631    9637
                9658    9726    9732    9739    9743    9747    9754    9758    9842    9931    9937    9950    9955    9976    9983
                9988   10135   10236   10243   10250   10259   10264   10269   10403   10437   10448   10458   10643   10709   10729
               10739   10798   10800   10819   10857   10884   10890   10901   10907   10913   10917   10923   10929   10948   10949
               11085   11154   11176   11201   11216   11221   11223   11236   11280   11297   11348   11363   11645   11655   11680
               11698   11711   11722   11732   11824   11906   11946   11952   11955   11968   11971   12021   12038   12077   12094
               12104   12113   12120   12176   12245   12309   12322   12387   12411   12418   12456   12471   12492   12560   12584
CLC             3365    9812   10651   11350   11365   11646   11656
CLR             2829    2836    2849    2850    3009    3029    3067    3069    3077    3078    3135    3552    3610    3618    4027
                4046    4078    4082    4095    4103    4144    4191    4229    4544    4549    4550    4552    4562    4563    4644
                4648    4865    5135    5257    5319    6008    6027    6244    6245    6246    6260    6263    6271    6273    6288
                6289    6297    6338    6348    6349    6364    6365    6379    6381    6395    6396    6408    6465    6466    6482
                6483    6496    6498    6513    6514    6526    6583    6584    6585    6599    6602    6610    6612    6678    6692
                6709    6714    6800    6814    6831    6836    6921    6935    6952    6957    7038    7039    7064    7133    7134
                7136    7153    7155    7247    7248    7312    7313    7333    7443    7641    7642    7644    7661    7664    7675
                7753    7754    7756    7774    7777    7883    7884    7922    8000    8014    8031    8036    8132    8135    8136
                8152    8154    8157    8231    8350    8449    8534    8763    8831    9052    9053    9078    9158    9159    9182
                9239    9321    9322    9336    9338    9429    9430    9444    9446    9552    9642    9645    9671    9672    9724
                9910   10074   10075   10077   10094   10162   10163   10165   10184   10225   10226   10237   10244   10251   10308
               10364   10490   10521   10704   10790   10847   10848   10849   10858   11165   11198   11215   11226   11281   11298
               11349   11364   11434   11667   11677   11729   11891   11932   11933   11937   11941   11970   11984   12013   12016
               12146   12179   12215   12216   12293   12294   12380   12469   12566
CLRB            3038    4304    4553    7131    7639    7751    8133    8134   11969   12042   12093   12119   12250   12318
CMP             2837    2860    2948    3085    3087    3358    3379    3558    3571    3578    3635    3637    3718    3734    3745
                3759    3771    3781    3791    3801    3811    3821    3831    3850    3861    3872    3884    3897    3908    3961
                3969    3981    3990    4021    4036    4044    4052    4091    4104    4124    4147    4158    4185    4195    4205
                4218    4269    4278    4308    4325    4356    4364    4392    4400    4891    4902    4913    4926    4945    4955
                4965    4975    4985    4995    5005    5024    5035    5044    5055    5064    5075    5138    6052    6123    7080
                7093    7186    7206    7209    7293    7427    7541    7551    7560    7588    7596    7690    7710    7713    7784
                7795    7802    7895    7908    7914    8091    8171    8188    8195    8234    8259    8268    9092    9105    9111
                9247    9260    9656    9836    9839    9843    9923    9944   10126   10137   10140   10427   10430   10440   10449
               10452   10461   10531   10734   10740   10803   10805   11206   11692   11704   11716   11737   11953   11977   12034
               12164   12172   12177   12185   12190   12192   12197   12225   12238
CMPB            3840    5015    7532   11959   11963   12086   12088   12096   12117   12121   12242   12248   12297   12299
COM             8767    8835    8905    8970    9817    9828    9830    9833    9835   10641   10660   10669   10677   10685
DEC             2944    3034    3074    3112    3116    3369    3391    3616    3624    3643    4079    4086    4099    4127    4708
                4769    4776    4871    5518    5670    5807    6078    6080    6082    6107    6109    6111    6144    6154    6235
                6261    6339    6368    6455    6486    6574    6600    6672    6679    6743    6794    6801    6865    6915    6922
                6986    7097    7213    7570    7717    7994    8001    8065    8190    8238    9107    9263    9325    9433    9541
                9652    9675    9785    9877    9878   10016   10021   10056   10142   10271   10273   10304   10310   10796   10860
               10862   10898   11160   11162   11167   11203   11228   11308   11310   11372   11374   11413   11426   11432   11438
               11445   11659   11661   11681   11712   11733   11829   11832   11846   11848   11850   11854   11887   11895   12222
```

```
        12388
DECB    12100  12103  12477  12488
EMT      1786
HALT     1899  10932  11911  12076  12361  12559  12583
INC      3033   3110   3111   3628   3630   3645   4085   4126   4145   4194   4266   4303   4353   4389   4870
         5512   8365   9646   9673  10124  10799  10802  11289  11306  11355  11370  11576  12020  12189  12196
        12224  12350  12483  12491  12567
INCB    11981  12123  12344
IOT      1787
JMP      1903   1905   2878   2964   3041   3044   3062   3955   4016   4075   4142   4180   7741   9651   9660
         9693  11670  12169
JSR      2874   2988   3020   3134   3178   3221   3247   3273   3300   3326   3349   3414   3440   3466   3492
         3518   3547   3606   3647   3691   3736   3747   3761   3773   3783   3793   3803   3813   3823   3833
         3842   3852   3863   3874   3886   3899   3910   3956   3960   3980   4017   4020   4035   4081   4090
         4143   4150   4181   4184   4201   4204   4263   4300   4349   4385   4444   4451   4465   4478   4516
         4533   4540   4566   4584   4620   4635   4652   4665   4701   4725   4738   4758   4793   4806   4844
         4893   4904   4915   4928   4947   4957   4967   4977   4987   4997   5007   5017   5026   5037   5046
         5057   5066   5077   5106   5127   5147   5165   5180   5217   5230   5250   5259   5277   5291   5324
         5336   5367   5376   5402   5416   5447   5459   5496   5521   5530   5546   5560   5593   5605   5641
         5657   5682   5695   5727   5739   5777   5794   5819   5832   5864   5876   5912   5922   5939   5952
         5977   5979   6003   6013   6021   6173   6186   6205   6207   6237   6248   6267   6274   6287   6291
         6301   6341   6351   6374   6382   6394   6398   6402   6412   6457   6468   6492   6499   6512   6516
         6520   6530   6576   6587   6606   6618   6693   6699   6711   6715   6732   6748   6815   6821   6833
         6837   6854   6870   6936   6942   6954   6958   6975   6991   7029   7042   7050   7054   7062   7065
         7071   7118   7123   7137   7143   7154   7156   7170   7175   7249   7261   7268   7280   7285   7292
         7315   7327   7335   7392   7412   7417   7426   7457   7464   7509   7531   7550   7576   7580   7626
         7631   7645   7651   7662   7665   7676   7742   7757   7763   7775   7778   7842   7857   7877   7885
         7894   7936   7943   8015   8021   8033   8037   8054   8070   8089   8127   8139   8145   8149   8155
         8158   8164   8229   8244   8248   8258   8282   8293   8335   8354   8378   8391   8407   8417   8419
         8452   8455   8470   8480   8501   8503   8537   8540   8555   8565   8584   8586   8623   8628   8637
         8647   8668   8670   8704   8709   8718   8728   8766   8771   8786   8796   8834   8839   8854   8864
         8903   8910   8919   8929   8968   8975   8984   8994   9015   9017   9043   9056   9064   9068   9076
         9079   9085   9148   9149   9162   9168   9172   9180   9185   9191   9210   9223   9242   9276   9278
         9304   9331   9341   9361   9374   9391   9412   9439   9454   9473   9486   9503   9525   9547   9562
         9581   9594   9611   9688   9810   9811   9825   9848   9922   9969  10078  10084  10095  10098  10134
        10136  10167  10173  10185  10187  10321  10344  10347  10365  10373  10385  10411  10497  10529  10737
        10743  10811  10818  10859  10943  11027  11033  11082  11086  11166  11170  11178  11180  11182  11184
        11186  11194  11205  11227  11231  11290  11307  11356  11371  11424  11430  11436  11440  11443  11513
        11522  11593  11673  11689  11702  11714  11725  11735  11886  11894  12095  12102  12109  12167  12356
MOV      2827   2835   2839   2841   2842   2843   2844   2845   2846   2847   2848   2852   2853   2856   2857
         2858   2859   2864   2866   2867   2868   2872   2873   2912   2923   2924   2933   2934   2936   2938
         2940   2941   2942   2943   2947   2949   2961   2965   2967   2980   2981   2984   2986   2987   2998
         3000   3002   3003   3004   3010   3018   3027   3028   3030   3031   3032   3068   3071   3072   3073
         3076   3079   3080   3084   3093   3099   3106   3108   3114   3115   3122   3132   3133   3136   3142
         3149   3156   3161   3162   3176   3213   3216   3219   3220   3239   3242   3245   3246   3265   3268
         3271   3272   3292   3295   3298   3299   3318   3321   3324   3325   3347   3350   3351   3352   3353
         3355   3357   3360   3361   3375   3376   3377   3381   3382   3383   3406   3409   3412   3413   3432
         3435   3438   3439   3458   3461   3464   3465   3484   3487   3490   3491   3510   3513   3516   3517
         3543   3545   3557   3561   3562   3567   3569   3570   3573   3577   3581   3582   3601   3604   3613
         3614   3615   3619   3620   3626   3629   3633   3640   3642   3690   3693   3694   3695   3697   3698
         3699   3700   3701   3702   3703   3704   3705   3710   3711   3715   3716   3717   3726   3732   3733
         3743   3744   3755   3757   3758   3769   3770   3779   3780   3789   3790   3799   3800   3809   3810
         3819   3820   3829   3830   3838   3839   3848   3849   3857   3858   3859   3870   3871   3882   3883
         3895   3896   3906   3907   3930   3951   3953   3957   3958   3963   3972   3978   3984   3988   3994
         4012   4014   4018   4023   4028   4029   4033   4038   4042   4043   4048   4049   4056   4057   4058
         4073   4076   4077   4083   4084   4088   4093   4096   4097   4098   4101   4102   4106   4107   4108
```

```
4122  4140  4146  4154  4156  4157  4162  4178  4182  4187  4190  4192  4197  4202  4207
4210  4211  4212  4213  4214  4215  4220  4221  4222  4228  4230  4239  4241  4242  4243
4260  4262  4264  4265  4268  4271  4272  4277  4282  4283  4298  4302  4305  4307  4311
4313  4316  4321  4323  4324  4330  4346  4348  4350  4351  4352  4355  4358  4359  4363
4368  4369  4381  4383  4386  4387  4388  4391  4394  4395  4399  4404  4405  4439  4441
4446  4448  4485  4487  4511  4513  4519  4520  4522  4523  4524  4525  4526  4527  4528
4529  4530  4545  4547  4548  4551  4554  4556  4561  4592  4594  4615  4617  4624  4625
4626  4627  4628  4629  4631  4632  4673  4675  4698  4702  4703  4704  4705  4706  4707
4710  4711  4746  4748  4757  4759  4764  4765  4766  4767  4771  4772  4773  4774  4775
4778  4779  4814  4816  4839  4842  4846  4847  4848  4850  4851  4852  4853  4854  4855
4856  4857  4858  4864  4866  4869  4874  4876  4879  4880  4882  4889  4890  4900  4901
4911  4912  4922  4924  4925  4940  4942  4943  4953  4954  4963  4964  4973  4974  4983
4984  4993  4994  5003  5004  5012  5014  5022  5023  5031  5032  5033  5042  5043  5053
5054  5062  5063  5073  5074  5096  5122  5124  5129  5136  5137  5140  5142  5143  5188
5190  5201  5202  5210  5211  5239  5241  5252  5256  5298  5300  5309  5310  5318  5345
5347  5363  5365  5369  5373  5396  5423  5425  5434  5435  5443  5468  5470  5491  5493
5498  5510  5511  5515  5516  5517  5523  5527  5528  5568  5570  5579  5580  5589  5614
5616  5636  5638  5643  5654  5667  5668  5669  5703  5705  5714  5715  5723  5748  5750
5772  5774  5779  5791  5804  5805  5806  5840  5842  5851  5852  5860  5885  5887  5907
5909  5914  5919  5934  5960  5962  5976  5978  5999  6001  6006  6009  6010  6030  6031
6033  6034  6043  6049  6050  6051  6055  6057  6059  6071  6072  6081  6098  6100  6102
6120  6122  6125  6127  6129  6134  6151  6167  6194  6196  6204  6206  6227  6230  6232
6233  6234  6242  6247  6254  6255  6256  6257  6258  6269  6272  6331  6334  6336  6337
6345  6347  6350  6357  6358  6359  6361  6363  6366  6367  6370  6377  6380  6447  6450
6452  6453  6454  6462  6464  6467  6474  6475  6477  6479  6481  6484  6485  6488  6494
6497  6566  6569  6571  6572  6573  6581  6586  6593  6594  6595  6596  6597  6608  6611
6613  6621  6658  6661  6668  6669  6670  6671  6675  6685  6689  6690  6691  6700  6701
6704  6705  6713  6733  6734  6737  6738  6739  6740  6741  6742  6780  6783  6790  6791
6792  6793  6797  6807  6811  6812  6813  6822  6823  6826  6827  6835  6855  6856  6859
6860  6861  6862  6863  6864  6901  6904  6911  6912  6913  6914  6918  6928  6932  6933
6934  6943  6944  6947  6948  6956  6976  6977  6980  6981  6982  6983  6984  6985  7024
7027  7034  7040  7041  7055  7056  7057  7059  7060  7061  7063  7076  7077  7078  7079
7082  7083  7091  7113  7116  7130  7135  7144  7145  7146  7149  7150  7182  7183  7184
7185  7188  7189  7207  7237  7240  7244  7245  7246  7254  7255  7256  7273  7274  7275
7290  7309  7310  7311  7320  7321  7322  7373  7376  7380  7381  7382  7385  7386  7387
7399  7400  7401  7405  7406  7407  7424  7444  7445  7446  7450  7451  7452  7504  7507
7510  7512  7513  7514  7515  7516  7517  7520  7521  7522  7524  7525  7526  7527  7529
7530  7539  7549  7553  7554  7555  7558  7577  7578  7579  7586  7621  7624  7638  7643
7652  7653  7654  7657  7658  7682  7686  7687  7688  7689  7692  7693  7711  7736  7739
7750  7755  7764  7765  7766  7770  7771  7782  7786  7787  7791  7793  7794  7797  7800
7804  7805  7837  7840  7845  7846  7847  7850  7851  7852  7864  7865  7866  7870  7871
7872  7892  7923  7924  7925  7929  7930  7931  7980  7983  7990  7991  7992  7993  7997
8007  8011  8012  8013  8022  8023  8026  8027  8035  8055  8056  8059  8060  8061  8062
8063  8064  8119  8122  8125  8137  8138  8150  8151  8153  8156  8167  8168  8169  8170
8173  8174  8186  8223  8226  8230  8232  8233  8242  8245  8246  8247  8251  8261  8266
8275  8289  8298  8330  8333  8336  8337  8339  8343  8345  8346  8347  8349  8364  8366
8398  8400  8416  8418  8443  8446  8462  8463  8464  8465  8466  8500  8502  8528  8531
8547  8548  8549  8550  8551  8583  8585  8613  8616  8619  8667  8669  8694  8697  8700
8757  8760  8778  8779  8780  8781  8782  8825  8828  8846  8847  8848  8849  8850  8893
8896  8899  8958  8961  8964  9014  9016  9038  9041  9048  9054  9055  9069  9070  9071
9073  9074  9075  9077  9088  9089  9090  9091  9094  9095  9103  9109  9113  9114  9115
9121  9144  9146  9154  9160  9161  9173  9174  9175  9177  9178  9179  9181  9194  9195
9200  9201  9203  9204  9231  9233  9243  9244  9245  9246  9249  9250  9258  9275  9277
9300  9302  9307  9314  9315  9316  9318  9320  9323  9324  9327  9334  9337  9348  9349
9354  9355  9356  9382  9384  9408  9410  9415  9422  9423  9424  9426  9428  9431  9432
9435  9442  9445  9461  9462  9466  9467  9468  9494  9496  9521  9523  9528  9535  9536
```

```
                9537    9539    9540    9543    9550    9553    9554    9569    9570    9574    9575    9576    9602    9604    9628
                9629    9634    9640    9648    9654    9655    9659    9678    9682    9685    9729    9735    9752    9762    9763
                9778    9779    9780    9781    9782    9783    9784    9787    9788    9789    9804    9805    9806    9807    9809
                9818    9824    9827    9831    9832    9838    9845    9868    9869    9870    9871    9872    9873    9876    9880
                9881    9882    9889    9903    9904    9905    9906    9908    9909    9920    9928    9934    9938    9942    9948
                9953    9956    9967    9973    9980    9986    9991   10005   10006   10007   10009   10010   10011   10012   10013
               10018   10023   10049   10050   10051   10052   10053   10054   10055   10058   10059   10060   10071   10076   10086
               10087   10090   10091   10096   10112   10113   10114   10115   10116   10117   10118   10119   10120   10121   10122
               10123   10125   10129   10130   10138   10145   10146   10147   10148   10149   10159   10164   10176   10177   10180
               10181   10186   10219   10220   10221   10222   10223   10224   10227   10228   10229   10230   10235   10242   10249
               10275   10276   10277   10278   10279   10280   10297   10298   10299   10300   10301   10302   10303   10306   10315
               10318   10319   10320   10324   10325   10326   10342   10349   10350   10351   10356   10357   10359   10360   10362
               10429   10433   10434   10439   10444   10445   10451   10454   10455   10459   10464   10465   10481   10482   10483
               10485   10486   10487   10489   10491   10495   10499   10500   10516   10517   10519   10520   10522   10523   10530
               10533   10534   10538   10627   10628   10629   10630   10631   10632   10633   10634   10639   10642   10647   10649
               10658   10661   10667   10670   10675   10678   10683   10686   10716   10717   10732   10764   10765   10766   10767
               10768   10769   10784   10786   10787   10788   10789   10827   10841   10842   10843   10844   10845   10846   10850
               10851   10852   10853   10856   10864   10865   10866   10867   10868   10869   10870   10871   10887   10894   10895
               10904   10911   10920   10926   10944   11014   11016   11017   11018   11019   11020   11021   11022   11024   11026
               11029   11030   11031   11032   11090   11091   11092   11093   11094   11095   11142   11143   11144   11145   11146
               11147   11148   11149   11150   11155   11157   11164   11169   11173   11174   11175   11177   11179   11181   11183
               11185   11191   11192   11193   11199   11200   11202   11217   11218   11219   11220   11222   11225   11230   11239
               11240   11241   11243   11267   11268   11269   11273   11279   11282   11285   11286   11296   11299   11302   11303
               11312   11335   11336   11337   11338   11339   11340   11341   11342   11345   11352   11360   11367   11376   11377
               11378   11379   11380   11402   11403   11404   11405   11406   11407   11408   11409   11410   11411   11412   11415
               11416   11418   11419   11422   11423   11425   11431   11435   11437   11441   11442   11444   11447   11448   11449
               11450   11451   11500   11502   11503   11504   11505   11506   11507   11508   11510   11512   11515   11516   11517
               11518   11519   11520   11521   11597   11598   11599   11600   11601   11602   11629   11630   11631   11632   11633
               11634   11635   11636   11637   11638   11640   11648   11649   11658   11663   11666   11668   11669   11671   11672
               11674   11678   11679   11691   11694   11695   11696   11697   11699   11700   11701   11703   11708   11709   11710
               11715   11718   11719   11720   11721   11723   11724   11726   11730   11731   11736   11739   11740   11741   11742
               11744   11787   11788   11789   11790   11791   11792   11793   11794   11795   11796   11805   11820   11821   11831
               11839   11840   11841   11858   11859   11860   11861   11862   11863   11876   11877   11878   11881   11882   11884
               11885   11892   11893   11898   11910   11948   11949   11951   11954   11967   11979   11980   11982   11983   11986
               11987   12003   12004   12005   12006   12007   12008   12009   12014   12017   12037   12043   12044   12045   12046
               12047   12049   12050   12078   12079   12083   12098   12147   12148   12149   12150   12152   12187   12199   12213
               12214   12217   12227   12236   12237   12252   12253   12254   12255   12285   12286   12287   12288   12289   12291
               12292   12311   12312   12313   12314   12315   12340   12346   12351   12364   12367   12379   12384   12393   12398
               12403   12404   12406   12409   12413   12420   12421   12452   12460   12461   12462   12468   12475   12493   12494
               12495   12496   12497   12512   12513   12517   12548   12549   12550   12551   12552   12553   12554   12555   12556
               12557   12558   12564   12565   12569   12570   12571   12572   12573   12574   12575   12576   12577   12580
MOVB            2851    3036    3137    3631    4267    4276    4320    4354    4362    4390    4398    4473    4574    4580    4660
                4733    4801    5175    5226    5286    5332    5411    5456    5555    5601    5690    5735    5827    5872    5947
                6181    6687    6688    6702    6703    6735    6736    6809    6810    6824    6825    6857    6858    6930    6931
                6945    6946    6978    6979    7036    7037    7058    7132    7147    7148    7640    7655    7656    7752    7768
                7769    8009    8010    8024    8025    8057    8058    8386    9050    9051    9072    9156    9157    9176    9218
                9369    9481    9589   10072   10073   10088   10089   10160   10161   10178   10179   10316   10317   10352   10354
               11985   12012   12015   12029   12032   12041   12080   12108   12116   12162   12183   12195   12223   12241   12246
               12295   12353   12453   12454   12457   12458   12459   12463   12466   12467   12486   12515
NEG            12011   12464
NOP             6038    6041    6140    6615    7353    7484    7963    8304    8490    8575    8657    8738    8806    8874    8939
                9004    9689    9690    9691    9849   10525   10526   10527   10528   11029   11040   11050   11061   11515   11529
               11537   11549   11561   11568   11579
RESET           3019    4539    9687    9823
ROL             3366    3388    7258    7277    7324    7383    7389    7402    7409    7447    7454    7848    7854    7867    7874
                7926    7933    9813   12302   12304   12306   12470   12472   12473   12474   12476
```

DERPSB.P11 CROSS REFERENCE TABLE -- PERMANENT SYMBOLS

```
ROR      3081  4867 10231 10232 10254 10255 10652 10653 10697 10698 10713 10714 10854 11270 11283
        11300 11347 11351 11362 11366 11644 11647 11654 11657
RTI      2865  9764  9890 10025 11988 12051 12085 12188 12200 12218 12228 12256 12316 12369 12498
        12582
RTS      2992  3225  3251  3277  3304  3330  3418  3444  3470  3496  3522  3655  3657  3935  4492
         4599  4678  4751  4819  5101  5195  5246  5304  5350  5430  5474  5574  5621  5709  5755
         5846  5892  5966  6198  6311  6317  6422  6432  6540  6550  6758  6762  6880  6884  7001
         7005  7345  7351  7476  7482  7955  7961  8078  8082  8402  9236  9387  9499  9607  9790
         9816  9850  9883  9911  9960  9993 10061 10101 10150 10190 10281 10327 10393 10401 10419
        10424 10469 10501 10539 10770 10828 10872 11096 11244 11313 11381 11452 11603 11745 11864
        11899 12125 12153 12423 12518
SEC      3387  7257  7276  7323  7388  7408  7453  7853  7873  7932 11346 11361 11643 11653
SUB      3641  6056  6058  6126  6128  7084  7190  7694  8175  9096  9251  9874  9921  9968 10008
        10131 10343 10518 10785 11015 11197 11417 11501 11676 11707 11728 12018 12352
TRAP    12520 12529 12530 12531 12532 12535 12536 12537 12538 12539 12540
TST      2875  2913  3118  3159  3160  3621  3916  3921  3934  4109  4120  4129  4225  4938  5083
         5088  5100  5103  5388  6044  6283  6390  6508  6674  6728  6796  6850  6917  6971  7072
         7085  7090  7166  7191  7569  7695  7996  8050  8165  8176  8185  8249  9086  9097  9102
         9252  9257  9643  9814 10132 10256 10261 10266 10374 10376 10425 10635 10654 10718 10794
        10945 11073 11075 11077 11187 11189 11195 11208 11213 11232 11275 11287 11292 11304 11353
        11368 11420 11428 11591 11664 11687 11806 11879 11889 11950 11974 12023 12033 12082 12090
        12112 12151 12168 12175 12180 12220 12310 12317 12359 12365 12415 12481 12514
TSTB     3005  3039  7581  9821 10353 10355 11961 12025 12039 12074 12114 12181 12407
.ASCII   2005  2006 12646 12651 12658 12666 12670 12675 12709 12724 12739 12748 12755 12789 12800
        12810 12824 12853 12868 12882 12901 12926 12940 12960 12978 12992 13003 13019 13039 13056
        13069 13091 13117 13141 13159 13181 13198
.ASCIZ   2004  2007  2883  2887  2891  2895  2899  2903  2907  2911  2920  2956  2960  3026  3049
         3053  3057  3061  3066  3092  3098  3104  3141  3148  3155  3186  3190  3194  3200  5395
         9633  9639  9696  9728  9734  9741  9745  9749  9756  9760 10886 10892 10903 10909 10915
        10919 10925 10931 11908 12260 12261 12262 12263 12265 12424 12586 12598 12607 12611 12617
        12622 12625 12632 12635 12639 12642 12686 12693 12703 12720 12729 12764 12772 12779 12814
        12836 12844 12861 12875 12892 12912 12923 12933 12950 12969 12985 12998 13011 13029 13048
        13063 13080 13104 13129 13150 13170 13190 13211
.BLKB   12134 12259
.BLKW    2741  2813  2814 11475 11476 11477 11478 11479 11480 11482 11483 11484 11485 11486 12056
.BYTE    1960  1961  1966  1967  1975  1976  1984  1985  1986  1987  2822  9695 12257 12258 12499
        12500 12501 12502 13284 13286 13288 13290 13293 13296 13297 13299 13302 13305 13307 13309
        13311 13314 13316 13318 13321 13324 13327 13330 13333 13335
.DSABL  12202
.ENABL      1  1758 12130
.END    13340
.ENDC    1763  1776  1778  1779  1780  1786  1878  1892  1904  1922  1933  1944  1946  1954  1958
         1960  1988  1996  2002  2003  2004  2005  2009  2464  2469  2503  2505  2833  2839  2840
         2843  2845  2847  2849  2850  2852  2854  2870  2883  2887  2891  2895  2899  2903  2907
         2911  2920  2929  2930  2931  2932  2933  2934  2956  2960  2973  2974  2978  2979  2980
         2981  2995  2996  2997  2998  2999  3015  3016  3017  3018  3019  3026  3049  3053  3057
         3061  3066  3092  3098  3104  3125  3126  3130  3131  3132  3133  3134  3141  3148  3155
         3167  3168  3173  3174  3175  3186  3190  3194  3200  3205  3206  3209  3210  3211  3231
         3232  3235  3236  3237  3257  3258  3261  3262  3263  3284  3285  3288  3289  3290  3310
         3311  3314  3315  3316  3339  3340  3343  3344  3345  3398  3399  3402  3403  3404  3424
         3425  3428  3429  3430  3450  3451  3454  3455  3456  3476  3477  3480  3481  3482  3502
         3503  3506  3507  3508  3538  3539  3541  3542  3543  3594  3595  3599  3600  3601  3663
         3677  3735  3746  3772  3782  3792  3802  3812  3822  3832  3851  3898  3909  3941  3942
         3949  3950  3951  3998  3999  4010  4011  4012  4064  4065  4070  4071  4072  4133  4134
         4137  4138  4139  4168  4169  4174  4175  4176  4250  4251  4254  4255  4256  4291  4292
         4295  4296  4297  4338  4339  4343  4344  4345  4375  4376  4379  4380  4381  4412  4422
```

```
          4430    4431    4437    4438    4439    4503    4504    4509    4510    4511    4606    4607    4613    4614    4615
          4689    4690    4695    4696    4697    4829    4830    4837    4838    4839    4892    4903    4914    4956    4966
          4976    4986    4996    5006    5025    5065    5076    5113    5114    5120    5121    5122    5355    5356    5361
          5362    5363    5395    5482    5483    5489    5490    5491    5627    5628    5634    5635    5636    5763    5764
          5770    5771    5772    5898    5899    5905    5906    5907    5973    5974    5975    5976    5988    5989    5997
          5998    5999    6201    6202    6203    6204    6217    6218    6225    6226    6227    6321    6322    6329    6330
          6331    6438    6439    6445    6446    6447    6557    6558    6564    6565    6566    6628    6645    6650    6651
          6656    6657    6658    6735    6757    6772    6773    6778    6779    6780    6857    6879    6893    6894    6899
          6900    6901    6978    7000    7014    7015    7021    7022    7023    7076    7097    7104    7105    7111    7112
          7113    7222    7223    7229    7230    7231    7356    7357    7364    7365    7366    7490    7491    7501    7502
          7503    7611    7612    7619    7620    7621    7726    7727    7734    7735    7736    7818    7819    7828    7829
          7830    7969    7970    7976    7977    7978    8057    8077    8107    8108    8114    8115    8116    8214    8215
          8219    8220    8221    8310    8311    8326    8327    8328    8413    8414    8415    8416    8428    8429    8439
          8440    8441    8453    8490    8497    8498    8499    8500    8513    8514    8524    8525    8526    8538    8575
          8580    8581    8582    8583    8598    8599    8609    8610    8611    8625    8657    8664    8665    8666    8667
          8679    8680    8690    8691    8692    8706    8738    8742    8743    8753    8754    8755    8769    8806    8810
          8811    8821    8822    8823    8837    8874    8878    8879    8889    8890    8891    6907    8939    8943    8944
          8954    8955    8956    8972    9004    9011    9012    9013    9014    9028    9029    9035    9036    9037    9087
          9088    9106    9107    9130    9131    9142    9143    9144    9272    9273    9274    9275    9289    9290    9298
          9299    9300    9395    9396    9406    9407    9408    9508    9509    9519    9520    9521    9618    9619    9620
          9626    9627    9628    9629    9633    9639    9664    9665    9666    9668    9671    9677    9680    9681    9685
          9687    9693    9695    9696    9699    9728    9734    9741    9745    9749    9756    9760   10886   10892   10903
         10909   10915   10919   10925   10931   10952   10953   11098   11120   11127   11129   11254   11256   11321   11323
         11389   11391   11462   11468   11494   11496   11611   11613   11755   11757   11868   11870   11908   11920   11923
         11928   11943   11945   11956   11959   11960   11961   11963   11965   11972   11976   11981   11982   11986   11989
         11990   11993   12060   12080   12130   12161   12170   12201   12202   12206   12217   12229   12230   12237   12239
         12242   12244   12260   12261   12267   12274   12280   12324   12327   12330   12344   12351   12356   12357   12358
         12359   12369   12370   12373   12388   12426   12430   12507   12513   12516   12528   12529   12530   12531   12532
         12533   12534   12535   12536   12537   12538   12539   12540   12547   12556   12557   12563   12569   12570   12580
         12582   12589
.EQUIV   1786    1787    1795    1810    1811    1840    1841    1842    1843    1844    1845    1846    1847    1848    1849
         1868    1869    1870    1871    1872    1873    1874    1875    1876    1877
.EVEN    2883    2887    2891    2895    2899    2903    2907    2911    2920    2956    2960    3026    3049    3053    3057
         3061    3066    3092    3098    3104    3141    3148    3155    3186    3190    3194    3200    5395    9633    9639
         9728    9734    9741    9745    9749    9756    9760   10886   10892   10903   10909   10915   10919   10925   10931
        11908   12136   12425   12588   13223   13338
.IF      1759    1776    1777    1778    1779    1780    1784    1850    1878    1902    1922    1933    1944    1945    1953
         1957    1959    1988    1996    2002    2003    2004    2008    2009    2463    2468    2502    2504    2833    2834
         2839    2841    2843    2845    2847    2849    2850    2852    2870    2882    2886    2890    2894    2898    2902
         2906    2910    2919    2928    2930    2931    2933    2934    2955    2959    2972    2974    2978    2980    2981
         2994    2996    2998    2999    3014    3016    3018    3019    3025    3048    3052    3056    3060    3065    3091
         3097    3103    3124    3126    3130    3132    3133    3134    3140    3147    3154    3166    3168    3173    3175
         3185    3189    3193    3199    3204    3206    3209    3211    3230    3232    3235    3237    3256    3258    3261
         3263    3283    3285    3288    3290    3309    3311    3314    3316    3338    3340    3343    3345    3397    3399
         3402    3404    3423    3425    3428    3430    3449    3451    3454    3456    3475    3477    3480    3482    3501
         3503    3506    3508    3537    3539    3541    3543    3593    3595    3599    3601    3660    3674    3734    3735
         3745    3746    3771    3772    3781    3782    3791    3792    3801    3802    3811    3812    3821    3822    3831
         3832    3850    3851    3897    3898    3908    3909    3940    3942    3949    3951    3997    3999    4010    4012
         4063    4065    4070    4072    4132    4134    4137    4139    4167    4169    4174    4176    4249    4251    4254
         4256    4290    4292    4295    4297    4337    4339    4343    4345    4374    4376    4379    4381    4411    4421
         4429    4431    4437    4439    4502    4504    4509    4511    4605    4607    4613    4615    4688    4690    4695
         4697    4828    4830    4837    4839    4891    4892    4902    4903    4913    4914    4955    4956    4965    4966
         4975    4976    4985    4986    4995    4996    5005    5006    5024    5025    5064    5065    5075    5076    5112
         5114    5120    5122    5354    5356    5361    5363    5394    5481    5483    5489    5491    5626    5628    5634
         5636    5762    5764    5770    5772    5897    5899    5905    5907    5972    5974    5976    5987    5989    5997
         5999    6200    6202    6204    6216    6218    6225    6227    6320    6322    6329    6331    6437    6439    6445
```

```
                6447    6556    6558    6564    6566    6627    6644    6649    6651    6656    6658    6734    6754    6771    6773
                6778    6780    6856    6876    6892    6894    6899    6901    6977    6997    7013    7015    7021    7023    7073
                7076    7094    7097    7103    7105    7111    7113    7221    7223    7229    7231    7355    7357    7364    7366
                7489    7491    7501    7503    7610    7612    7619    7621    7725    7727    7734    7736    7817    7819    7828
                7830    7968    7970    7976    7978    8056    8076    8106    8108    8114    8116    8213    8215    8219    8221
                8309    8311    8326    8328    8412    8414    8416    8427    8429    8439    8441    8453    8489    8496    8498
                8500    8512    8514    8524    8526    8538    8574    8579    8581    8583    8597    8599    8609    8611    8625
                8656    8663    8665    8667    8678    8680    8690    8692    8706    8737    8741    8743    8753    8755    8767
                8805    8809    8811    8821    8823    8835    8873    8877    8879    8889    8891    8905    8938    8942    8944
                8954    8956    8970    9003    9010    9012    9014    9027    9029    9035    9037    9087    9106    9129    9131
                9142    9144    9271    9273    9275    9288    9290    9298    9300    9394    9396    9406    9408    9507    9509
                9519    9521    9617    9618    9620    9626    9628    9629    9632    9638    9663    9664    9665    9666    9667
                9668    9670    9676    9679    9681    9685    9687    9693    9695    9696    9727    9733    9740    9744    9748
                9755    9759   10885   10891   10902   10908   10914   10918   10924   10930   10951   10952   11097   11119   11126
               11128   11253   11255   11320   11322   11388   11390   11461   11467   11493   11495   11610   11612   11754   11756
               11867   11869   11907   11919   11922   11927   11932   11943   11955   11957   11958   11959   11961   11962   11963
               11972   11974   11982   11983   11988   11989   11990   11992   12059   12080   12129   12131   12159   12164   12170
               12202   12205   12206   12216   12229   12237   12238   12242   12243   12259   12260   12267   12273   12276   12292
               12326   12329   12339   12347   12354   12356   12357   12359   12362   12369   12370   12372   12387   12403   12429
               12506   12512   12516   12520   12529   12530   12531   12532   12533   12534   12535   12536   12537   12538   12539
               12540   12546   12556   12557   12562   12569   12570   12578   12580   12582   12586
.IFF            1776    1778    1779    1780    1786    1946    1954    1957    1959    1988    2009    2464    2469    2503    2505
                2839    2929    2930    2932    2933    2934    2973    2974    2979    2980    2981    2995    2996    2997    2998
                2999    3015    3016    3017    3018    3019    3125    3126    3131    3132    3133    3167    3168    3174    3175
                3205    3206    3210    3211    3231    3232    3236    3237    3257    3258    3262    3263    3284    3285    3289
                3290    3310    3311    3315    3316    3339    3340    3344    3345    3398    3399    3403    3404    3424    3425
                3429    3430    3450    3451    3455    3456    3476    3477    3481    3482    3502    3503    3507    3508    3538
                3539    3542    3543    3594    3595    3600    3601    3941    3942    3950    3951    3998    3999    4011    4012
                4064    4065    4071    4072    4133    4134    4138    4139    4168    4169    4175    4176    4250    4251    4255
                4256    4291    4292    4296    4297    4338    4339    4344    4345    4375    4376    4380    4381    4412    4422
                4430    4431    4438    4439    4503    4504    4510    4511    4606    4607    4614    4615    4689    4690    4696
                4697    4829    4830    4838    4839    5113    5114    5121    5122    5355    5356    5362    5363    5482    5483
                5490    5491    5627    5628    5635    5636    5763    5764    5771    5772    5898    5899    5906    5907    5973
                5974    5975    5976    5988    5989    5998    5999    6201    6202    6203    6204    6217    6218    6226    6227
                6321    6322    6330    6331    6438    6439    6446    6447    6557    6558    6565    6566    6628    6645    6650
                6651    6657    6658    6735    6757    6772    6773    6779    6780    6857    6879    6893    6894    6900    6901
                6978    7000    7014    7015    7022    7023    7104    7105    7112    7113    7222    7223    7230    7231    7356
                7357    7365    7366    7490    7491    7502    7503    7611    7612    7620    7621    7726    7727    7735    7736
                7818    7819    7829    7830    7969    7970    7977    7978    8056    8076    8107    8108    8115    8116    8214
                8215    8220    8221    8310    8311    8327    8328    8413    8414    8415    8416    8428    8429    8440    8441
                8489    8497    8498    8499    8500    8513    8514    8525    8526    8574    8580    8581    8582    8583    8598
                8599    8610    8611    8656    8664    8665    8666    8667    8679    8680    8691    8692    8737    8742    8743
                8754    8755    8806    8810    8811    8822    8823    8874    8878    8879    8890    8891    8939    8943    8944
                8955    8956    9004    9011    9012    9013    9014    9028    9029    9036    9037    9130    9131    9143    9144
                9272    9273    9274    9275    9289    9290    9299    9300    9395    9396    9407    9408    9508    9509    9520
                9521    9618    9619    9620    9627    9628    9629    9664    9667    9671    9677    9680    9695   10952   10953
               11098   11120   11127   11129   11254   11256   11321   11323   11389   11391   11462   11468   11494   11496   11611
               11613   11755   11757   11868   11870   11920   11956   11959   11960   11963   11989   11993   12060   12130   12202
               12206   12209   12217   12229   12230   12238   12243   12259   12274   12327   12329   12347   12369   12370   12373
               12387   12403   12430   12507   12513   12547   12563   12580
.IFT            2883    2887    2891    2895    2899    2903    2907    2911    2920    2956    2960    3026    3049    3053    3057
                3061    3066    3092    3098    3104    3141    3148    3155    3186    3190    3194    3200    5395    6734    6754
                6856    6876    6977    6997    8056    8076    8453    8489    8538    8574    8625    8656    8706    8737    8767
                8805    8835    8873    8905    8938    8970    9003    9633    9639    9728    9734    9741    9745    9749    9756
                9760   10886   10892   10903   10909   10915   10919   10925   10931   11908   11971   12208   12213   12297   12317
               12324   12357
```

DERPSB.P11 CROSS REFERENCE TABLE -- PERMANENT SYMBOLS

.IFTF 2883 2887 2891 2895 2899 2903 2907 2911 2920 2956 2960 3026 3049 3053 3057
3061 3066 3092 3098 3104 3141 3148 3155 3186 3190 3194 3200 5395 9633 9639
9720 9734 9741 9745 9749 9756 9760 10886 10892 10903 10909 10915 10919 10925 10931
11908 11969 12202 12206 12209 12293 12301 12323 12356
.IIF 1758 1763 1768 1773 1774 1775 1776 1779 1780 1781 1782 1899 2008 2840 2843
2849 2850 2852 2853 3150 3157 9665 9671 9672 9683 9695 9699 11923 11924 11925
11926 11927 11928 11932 11970 11971 11986 11989 11990 12127 12130 12136 12171 12252 12260
12267 12324 12330 12331 12332 12333 12334 12339 12362 12369 12370 12385 12410 12414 12528
12529 12530 12531 12532 12535 12536 12537 12538 12539 12540
.IRP 2833 2928 2972 2994 3014 3124 3166 3204 3230 3256 3283 3309 3338 3397 3423
3449 3475 3501 3537 3593 3940 3997 4063 4132 4167 4249 4290 4337 4374 4429
4502 4605 4688 4828 5112 5354 5481 5626 5762 5897 5972 5987 6200 6216 6320
6437 6556 6649 6771 6892 7013 7103 7221 7355 7489 7610 7725 7817 7968 8106
8213 8309 8412 8427 8496 8512 8579 8597 8663 8678 8741 8809 8877 8942 9010
9027 9129 9271 9288 9394 9507 9618 9778 9787 9868 9880 10049 10058 10112 10145
10219 10221 10276 10297 10324 10481 10499 10516 10538 10627 10764 10787 10827 10841 10866
11016 11090 11147 11243 11267 11312 11335 11377 11405 11447 11502 11597 11634 11744 11788
11858 11878 11898 11932 12003 12043 12287 12313 12339 12550 12556 12569 12570
.LIST 1 1758 1769 1779 1892 1899 1988 1990 1991 1992 1993 1994 1995 1996 1997
1998 1999 2000 2001 2002 2833 2854 2870 2883 2887 2891 2895 2899 2903 2907
2911 2920 2928 2933 2936 2956 2960 2972 2980 2984 2994 2998 3000 3014 3018
3026 3049 3053 3057 3061 3066 3092 3098 3104 3124 3132 3141 3148 3155 3166
3175 3176 3186 3190 3194 3200 3204 3211 3216 3230 3237 3242 3256 3263 3268
3283 3290 3295 3309 3316 3321 3338 3345 3347 3397 3404 3409 3423 3430 3435
3449 3456 3461 3475 3482 3487 3501 3508 3513 3537 3543 3545 3593 3601 3604
3659 3678 3940 3951 3953 3997 4012 4014 4063 4072 4073 4132 4139 4140 4167
4176 4178 4249 4256 4260 4290 4297 4298 4337 4345 4346 4374 4381 4383 4429
4439 4441 4502 4511 4513 4605 4615 4617 4688 4697 4698 4828 4839 4842 5112
5122 5124 5354 5363 5365 5395 5481 5491 5493 5626 5636 5638 5762 5772 5774
5897 5907 5909 5972 5976 5987 5999 6001 6200 6204 6216 6227 6230 6320 6331
6334 6437 6447 6450 6556 6566 6569 6649 6658 6661 6771 6780 6783 6892 6901
6904 7013 7023 7027 7103 7113 7116 7221 7231 7240 7355 7366 7376 7489 7503
7507 7610 7621 7624 7725 7736 7739 7817 7830 7840 7968 7978 7983 8106 8116
8122 8213 8221 8226 8309 8328 8333 8412 8416 8427 8441 8446 8496 8500 8512
8526 8531 8579 8583 8597 8611 8616 8663 8667 8678 8692 8697 8741 8755 8760
8809 8823 8828 8877 8891 8896 8942 8956 8961 9010 9014 9027 9037 9041 9129
9144 9146 9271 9275 9288 9300 9302 9394 9408 9410 9507 9521 9523 9618 9628
9633 9639 9671 9687 9728 9734 9741 9745 9749 9756 9760 10886 10892 10903 10909
10915 10919 10925 10931 11908 11927 12229 12369 12520 12528 12529 12530 12531 12532 12533
12535 12536 12537 12538 12539 12540 12541
.MACRO 1 1769 1780 1951 2928 2971 3124 3166 3204 3230 3256 3283 3309 3337 3397
3423 3449 3475 3501 3535 3592 3939 3997 4062 4131 4166 4247 4289 4335 4373
4428 4501 4604 4687 4827 5111 5353 5480 5625 5761 5896 5986 6215 6319 6436
6555 6648 6770 6891 7012 7102 7218 7354 7487 7609 7724 7816 7967 8105 8212
8308 8427 8596 9026 9128 9287 9393 9506 9618 12520
.MCALL 1758 1892 2854
.NLIST 1 1758 1769 1779 1892 1899 1988 1990 1991 1992 1993 1994 1995 1996 1997
1998 1999 2000 2001 2002 2833 2854 2870 2883 2887 2891 2895 2899 2903 2907
2911 2920 2928 2933 2936 2956 2960 2972 2980 2984 2994 2998 3000 3014 3018
3026 3049 3053 3057 3061 3066 3092 3098 3104 3124 3132 3141 3148 3155 3166
3175 3176 3186 3190 3194 3200 3204 3211 3216 3230 3237 3242 3256 3263 3268
3283 3290 3295 3309 3316 3321 3338 3345 3347 3397 3404 3409 3423 3430 3435
3449 3456 3461 3475 3482 3487 3501 3508 3513 3537 3543 3545 3593 3601 3604
3659 3678 3940 3951 3953 3997 4012 4014 4063 4072 4073 4132 4139 4140 4167
4176 4178 4249 4256 4260 4290 4297 4298 4337 4345 4346 4374 4381 4383 4429
4439 4441 4502 4511 4513 4605 4615 4617 4688 4697 4698 4828 4839 4842 5112

```
            5122    5124    5354    5363    5365    5395    5481    5491    5493    5626    5636    5638    5762    5772    5774
            5897    5907    5909    5972    5976    5987    5999    6001    6200    6204    6216    6227    6230    6320    6331
            6334    6437    6447    6450    6556    6566    6569    6649    6658    6661    6771    6780    6783    6892    6901
            6904    7013    7023    7027    7103    7113    7116    7221    7231    7240    7355    7366    7376    7489    7503
            7507    7610    7621    7624    7725    7736    7739    7817    7830    7840    7968    7978    7983    8106    8116
            8122    8213    8221    8226    8309    8328    8333    8412    8416    8427    8441    8446    8496    8500    8512
            8526    8531    8579    8583    8597    8611    8616    8663    8667    8678    8692    8697    8741    8755    8760
            8809    8823    8828    8877    8891    8896    8942    8956    8961    9010    9014    9027    9037    9041    9129
            9144    9146    9271    9275    9288    9300    9302    9394    9408    9410    9507    9521    9523    9618    9628
            9633    9639    9671    9687    9728    9734    9741    9745    9749    9756    9760   10886   10892   10903   10909
           10915   10919   10925   10931   11908   11927   12229   12369   12520   12528   12529   12530   12531   12532   12533
           12535   12536   12537   12538   12539   12540   12541
.PAGE       1951    2009    2462    2657    2710    2787    2826    5476    5758    9661    9884   10950   11005   11916   11990
           12324   12504   12544   12589
.REM           1
.REPT       1899    1990    1996    3660    3674
.SBTTL      1769    1782    1893    1902    1910    1951    2009    2660    2826    2833    2928    2972    2994    3014    3124
            3166    3204    3230    3256    3283    3309    3338    3397    3423    3449    3475    3501    3537    3593    3937
            3940    3997    4063    4132    4167    4249    4290    4337    4374    4429    4502    4605    4688    4828    5112
            5354    5481    5626    5762    5897    5972    5987    6200    6216    6320    6437    6556    6649    6771    6892
            7013    7103    7221    7355    7489    7610    7725    7817    7968    8106    8213    8309    8412    8427    8496
            8512    8579    8597    8663    8678    8741    8809    8877    8942    9010    9027    9129    9271    9288    9394
            9507    9618    9661    9702    9768    9796    9851    9884    9901    9914   10041   10066   10102   10192   10505
           10542   10878   10950   11917   11990   12057   12127   12271   12324   12370   12427   12504   12520   12544
.TITLE      1758
.WORD       1899    1900    1901    1959    1962    1963    1964    1965    1968    1969    1970    1971    1972    1973    1974
            1977    1978    1979    1988    1990    1991    1992    1993    1994    1995    1996    1997    1998    1999    2000
            2001    2742    2743    2745    2747    2748    2810    2989    2990    3222    3223    3248    3249    3274    3275
            3301    3302    3327    3328    3415    3416    3441    3442    3467    3468    3493    3494    3519    3520    4031
            4199    4714    4717    4782    4785    6294    6401    6519    6523    7030    7031    7032    7119    7120    7121
            7124    7125    7126    7250    7251    7252    7262    7263    7264    7269    7270    7271    7281    7282    7283
            7316    7317    7318    7328    7329    7330    7393    7394    7395    7413    7414    7415    7458    7459    7460
            7627    7628    7629    7632    7633    7634    7743    7744    7745    7858    7859    7860    7878    7879    7880
            7937    7938    7939    8128    8129    8130    8291    9044    9045    9046    9150    9151    9152    9676    9679
            9694   10498   11028   11034   11035   11036   11037   11038   11083   11084   11087   11088   11514   11594   11595
           12124   12131   12132   12133   12320   12323   12396   12401   12503   12579   12581   13224   13226   13228   13230
           13233   13236   13237   13239   13242   13245   13247   13249   13252   13255   13258   13261   13264   13268   13271
           13274   13277   13280
```

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

*,DERPSB.SEQ/SOL/CRF/PAGNUM/NL:TOC=SYSMAC.CO,DERPSB.P11
RUN-TIME:  85 128 18 SECONDS
RUN-TIME RATIO:  937/232=4.0
CORE USED:  38K  (75 PAGES)
```

Spooler runtime 42 Seconds, 175 KCS, 1213 disk reads, 4 disk writes, 294 pages