RH11-RP04

MASSBUS INTERFACE/DCL

MD-11-DZRPS-D

EP-DZRPS-D-DL-B

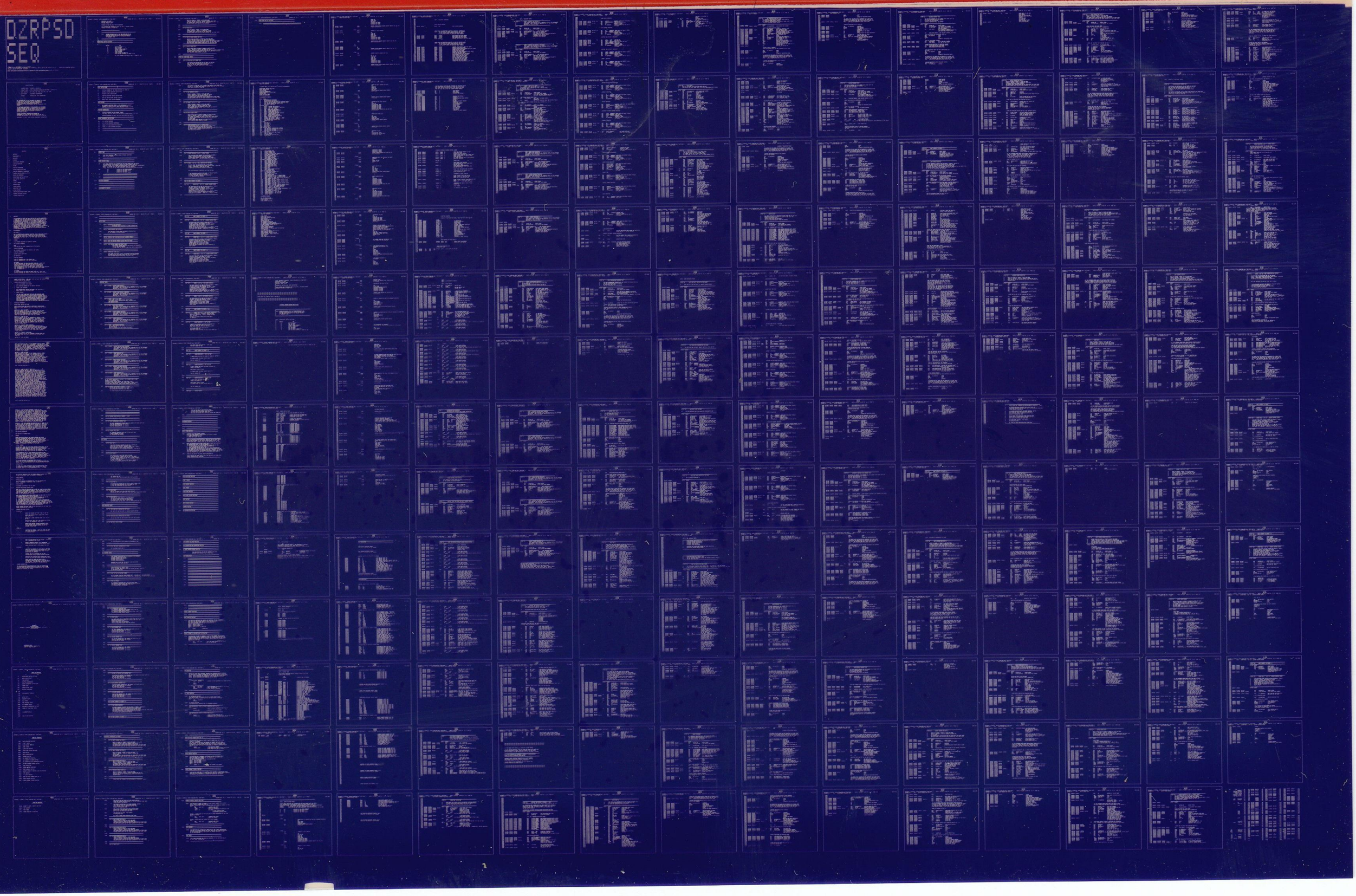
SEP 1975

Copyright © 1975

digital

Made In U.S.A.

FICHE 1 OF 2



RH11-RP04

MASSBUS INTERFACE/DCL

MD-11-DZRPS-D

EP-DZRPS-D-DL-B

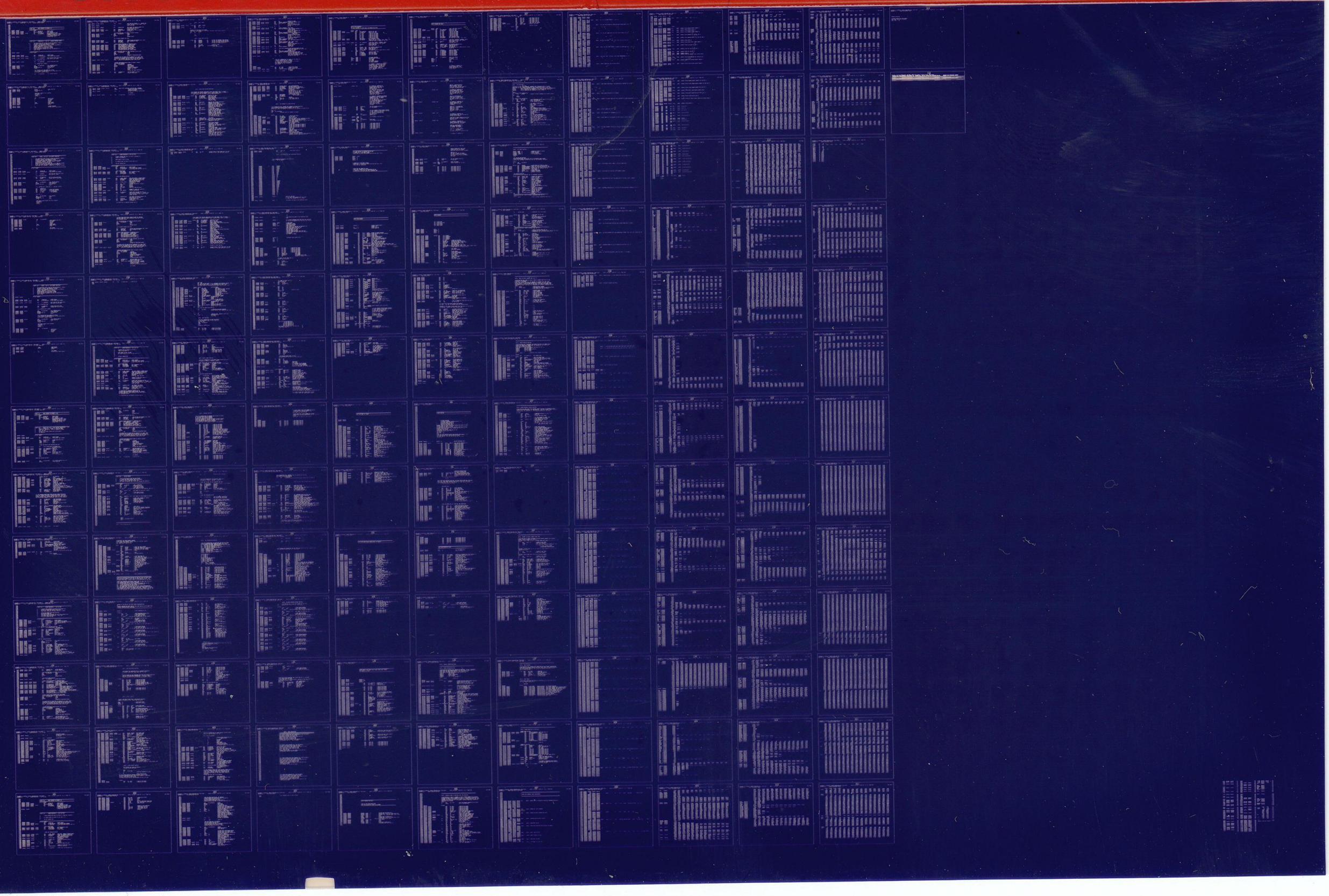
SEP 1975

Copyright © 1975

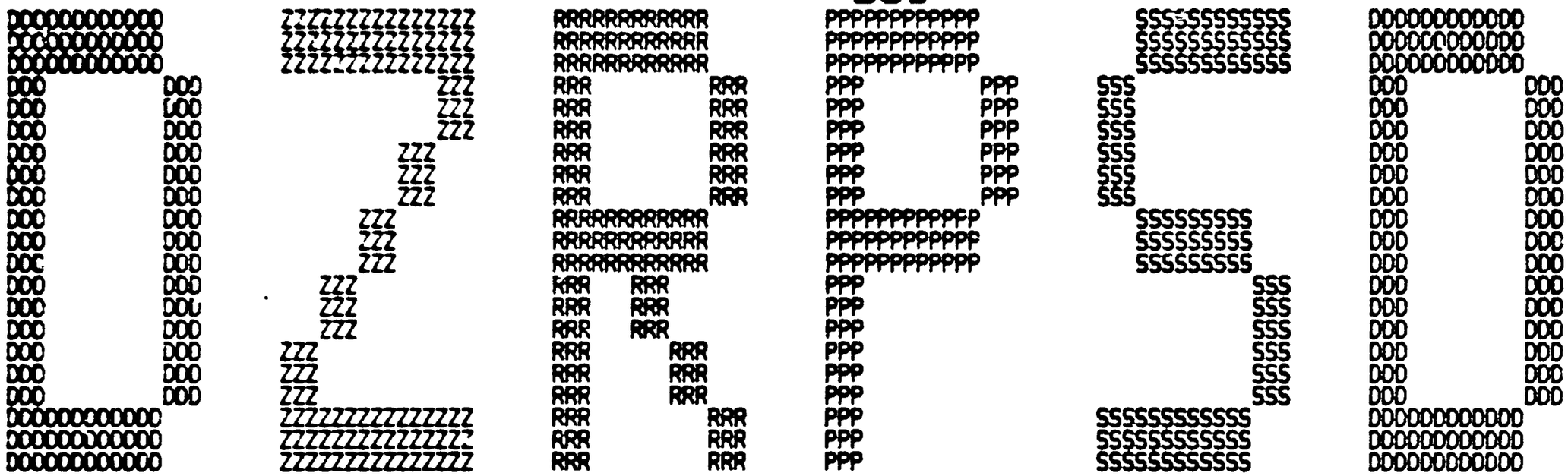
digital

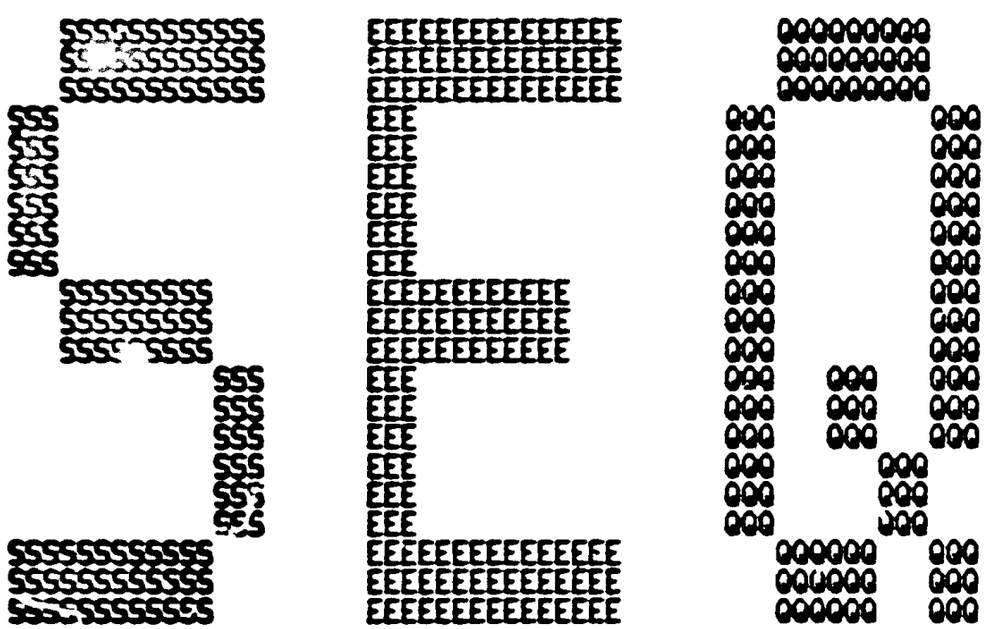
FICHE 2 OF 2

Made In U.S.A.



B01





LPTSPL Version 100(2025) Running on MTA260
 START User WELLINGHAM,A [400,3722] Job DZRPSD Seq. 38 Date 09-Oct-75 17:25:11 Monitor   RK213C KL10 SYS#1026 *START*
Request created: 09-Oct-75 17:19:38

[illegible] 09-Oct-75 17:25:11

C01

IDENTIFICATION

SEQ 0001

PRODUCT CODE: MAINDEC-11-DZRPS-D-D

PRODUCT NAME: RJP04 DISKLESS CONTROLLER TEST-PART I (STATIC 1A)

DATE CREATED: OCT 1974, MAR 1975, MAY 1975, AUG 1975

MAINTAINER: DIAGNOSTIC ENGINEERING

AUTHOR: SUB MALLICK, PETE BLACKSTONE

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

THE SOFTWARE DESCRIBED IN THIS DOCUMENT IS FURNISHED
TO THE PURCHASER UNDER A LICENSE FOR USE ON A SINGLE
COMPUTER SYSTEM AND CAN BE COPIED (WITH INCLUSION OF
DIGITAL'S COPYRIGHT NOTICE) ONLY FOR USE IN SUCH
SYSTEM, EXCEPT AS MAY OTHERWISE BE PROVIDED IN
WRITING BY DIGITAL.

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

COPYRIGHT (C) 1974, 1975 DIGITAL EQUIPMENT CORPORATION

D01

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
6.1 'FATAL' ERRORS
7. RESTRICTIONS
8. MISCELLANEOUS
8.1 EXECUTION TIME
8.2 STACK POINTER
8.3 OPERATOR SELECTABLE SCOPE LOOPS
8.4 PROGRAM REVISION HISTORY
9.0 PROGRAM DESCRIPTION

E01

1.0 ABSTRACT

THIS DIAGNOSTIC TESTS THE RH11 AND DCL OF AN RP04 SUBSYSTEM.
IT DOES NOT USE THE DISK SURFACE OR ANY SIGNALS FROM THE MOLI.
IT REQUIRES THAT THE DCL CABLE BE PLUGGED INTO THE MOLI 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 RH11 CONTROLLER,
A DISK CONTROL LOGIC (DCL), THE CABLE FROM THE DCL CAN BE
CONNECTED TO THE MOLI BUT IF NOT THAT CABLE MUST BE PROPERLY
TERMINATED.

2.2 STORAGE

THIS PROGRAM REQUIRES 16K WORDS OF MEMORY.

2.3 PRELIMINARY PROGRAMS

NONE
THIS CAN BE THE FIRST PROGRAM RUN ON THE SYSTEM.

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. F01

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 OFERATOR 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.

SEQ 0005

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.

6.1 'FATAL' ERRORS

IN THE EVENT THAT THE DISK DRIVE BECOMES UNAVAILABLE TO THE CONTROLLER, POWERS DOWN, OR CERTAIN CRITICAL STATUS BITS CANNOT BE CLEARED PRIOR TO THE START OF A TEST SEQUENCE - THIS INFORMATION WILL BE COMMUNICATED TO THE OPERATOR. IN ADDITION THE TTY BELL WILL RING AND THE PROGRAM WILL HALT.

IT IS SUGGESTED THAT IF THIS HAPPENS THE OPERATOR LOAD ADDRESS 200 (210) AND RESTART THE PROGRAM AS A FIRST ATTEMPT TO SOLVE THE PROBLEM. IF THE FAILURE CONTINUES TO OCCUR, LOOK IN THE TEST LISTING FOR THE 'HALT' INSTRUCTION AND REPLACE IT PLUS THE TWO WORDS ("TYPE ,CPHALT") ABOVE WITH 'NOP'S. WITH TTY ERROR PRINTOUTS INHIBITED A SCOPE LOOP CAN BE INITIATED FOR THE TEST IN QUESTION.

IT IS ALSO POSSIBLE TO CONTINUE FROM THE HALT POINT, BUT IT IS NOT RECOMMENDED AS ALL FOLLOWING TESTS WILL EXHIBIT THE SAME SYMPTOMS AND GIVE MISLEADING ERROR PRINTOUTS.

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 30 SECONDS PER DRIVE. SUBSEQUENT PASSES WILL TAKE 1 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. FOR INSTRUCTIONS REGARDING USAGE OF THIS TECHNIQUE, HIT ^C ANY TIME WHILE THE PROGRAM IS RUNNING. 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.

8.4 PROGRAM REVISION HISTORY

REVISION B:

1. MOVED THE UNIBUS INIT TEST (T36) TO BE T25.
2. MOVED THE COMMAND PARITY ERROR TEST (MCPE - T22) TO BE T11.
3. MOVED THE ILLEGAL FUNCTION TEST (ILF) T74 TO BE T37.
4. MODIFIED DVA, DPR, RDY & DRY STATUS BIT TESTS TO HALT THE PROGRAM IF THEY FAIL. SEQ 0008
5. ADDED DELAY BEFORE ISSUING DIAGNOSTIC MODE SECTOR PULSES (SP) TO ACCOUNT FOR THE SILO FILLING DELAY BETWEEN ISSUING GO AND THE RAISING OF RUN LINE.

REVISION C:

1. MODIFIED T23 (RHCS2 - 'NED' BIT TEST) SO THAT BOTH RHAS AND RHCS2 UNITS TABLES ARE PRINTED IN THEIR ENTIRETY.

2. MADE THE 2ND PORTION OF T23 (RHCS2 - 'CLR BIT) A SEPARATE TEST (T24). J01

3. ADDED A WARNING TYPEOUT TO T4 REGARDING OPERATOR VERIFICATION OF DRIVES BEING TESTED.

4. ADDED REVISION DATE TO THE HEADER TYPE OUT.

REVISION D:

1. MODIFIED THE PROGRAM SO THAT WITH A LOAD AND START AT LOCATION 210, THE PROGRAM WILL LOOP ON TESTING THE SELECTED DRIVE EVEN IF THAT DRIVE FAILS TO RESPOND.

2. ENABLED THE TTY INTERRUPT BIT IN THE NON-STANDARD DEVICE ADDRESS RESTART ROUTINE.

3. MODIFIED THE ATTENTION WITH ERROR TEST (T70) SO THAT ALL BITS ARE TESTED IN RHER2 AND RHER3.

4. ADDED TESTS OF RHMC FOR ZERO AFTER A READ/WRITE HEADER AND DATA TO T55,T56,T57 & T61,T62,T63. THIS WAS DONE TO VERIFIY PROPER COUNTDOWN OPERATION OF THE WORD COUNTER.

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-DZRPS-D, RJP04 DISKLESS RH11 TEST-PART 1
****

COPYRIGHT 1975
DIGITAL EQUIPMENT CORPORATION
MAYNARD, MASS. 01754

TABLE OF CONTENTS
****

43 OPERATIONAL SWITCH SETTINGS
58 BASIC DEFINITIONS
164 TRAP CATCHER
171 STARTING ADDRESS(ES)
184 MEMORY MANAGEMENT DEFINITIONS
224 COMMON TAGS
280 ERROR POINTER TABLE
964 REGISTER ADDRESSES
1122
1123 ***DIAGNOSTIC CODE***
1124
1126 SETUP TESTS
1525 REGISTER TESTS
2450 SILO TESTS
2742 MORE REGISTER TESTS
2889 DCL COMMAND TESTS
4592 READ/WRITE ADDRESSING VIA RHMR
5896 ERROR BIT FUNCTIONAL TESTS
7813
7814 ***SUBROUTINES***
7815
7820 END OF PASS ROUTINE

TABLE OF CONTENTS
****

7930 SAVE REGISTERS ROUTINE
7960 FLOAT 1 AND 0
8019 CLEAR MEMORY ROUTINE
8052 LOCAL TRAPS
8069 CLEAR DISK ROUTINE
8082 CHECK DISK STATUS ROUTINES
8212 SAVE ROUTINE
8240 WRITE CHECK ROUTINE
8284 COMPARE ROUTINE
8380 CRC GENERATION ROUTINE
8698 JAM CURRENT CYLINDER ROUTINE
8737 ECC GENERATION AND COMPARISON ROUTINE
9077 RH BASE ADDRESS CHANGE ROUTINE
9156 DISK SIMULATION
10168 SYSMAC LIBRARY ROUTINES
10172 SCOPE HANDLER ROUTINE
10246 CONVERT BINARY TO DECIMAL AND TYPE ROUTINE
10314 TYPE ROUTINE
10361 TTY INPUT ROUTINE
10464 READ AN OCTAL NUMBER FROM THE TTY
10518 ERROR HANDLER ROUTINE
10561 ERROR MESSAGE TYPEOUT ROUTINE

TABLE OF CONTENTS
****

10618 BINARY TO OCTAL (ASCII) AND TYPE
10696 TRAP DECODER
10711 TRAP TABLE
10732 POWER DOWN AND UP ROUTINES

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

                        PROGRAM BY SUB MALLICK, PETE BLACKSTONE

                        THIS PROGRAM WAS ASSEMBLED USING THE PDP-11 MAINDEC SYSMAC
                        PACKAGE (MAINDEC-11-DZQAC-A3).


                28      ********************************

                32      ****************************************************************

                        NOTE:   MACROS BEGINNING WITH ".$" ARE SUPPLIED BY AN
                                EXTERNAL SYSMAC.SML SYSTEM MACRO PACKAGE WHICH
                                MUST BE MADE AVAILABLE TO THE SOURCE PROGRAM
                                AT ASSEMBLY TIME.

                        ****************************************************************

                ************************************************************
        43      OPERATIONAL SWITCH SETTINGS
                ************************************************************

                44
                                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

        55                         6            ECC TEST-COMPARE END RESULTS ONLY IF SW08 IS LOW
```

```
                ;*****************************************************************
        58      ;BASIC DEFINITIONS
                ;*****************************************************************

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

                71      GENERAL PURPOSE REGISTER DEFINITIONS

                83      PRIORITY LEVEL DEFINITIONS

                93      "SWITCH REGISTER" SWITCH DEFINITIONS

                121     DATA BIT DEFINITIONS (BIT00 TO BIT15)

                149     BASIC "CPU" TRAP VECTOR ADDRESSES

                ;*****************************************************************
        164     ;TRAP CATCHER
                ;*****************************************************************

                167     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

                ;*****************************************************************
        171     ;STARTING ADDRESS(ES)
                ;*****************************************************************

                178     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

                ;*****************************************************************
        184     ;MEMORY MANAGEMENT DEFINITIONS
                ;*****************************************************************

                186     KT11 VECTOR ADDRESS

                190     KT11 STATUS REGISTER ADDRESSES

                197     KERNAL "I" PAGE DESCRIPTOR REGISTERS

                208     KERNAL "I" PAGE ADDRESS REGISTERS

                219     ;*****************************************************************

                222     ;*****************************************************************
```

```
                ;*****************************************************************
        224     ;COMMON TAGS
                ;*****************************************************************

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

                278     ;*****************************************************************

                ;*****************************************************************
        280     ;ERROR POINTER TABLE
                ;*****************************************************************

                282     ;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:

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

                723     ;*****************************************************************
                        ;*****************************************************************

                763     ;*****************************************************************
                        ;*****************************************************************

                ;*****************************************************************
        964     ;REGISTER ADDRESSES
                ;*****************************************************************

                ;*****************************************************************
        1122
                ;*****************************************************************

                ;*****************************************************************
        1123    ;***DIAGNOSTIC CODE***
                ;*****************************************************************
```

```
                ;*****************************************************************
1124
                ;*****************************************************************

                ;*****************************************************************
1126            ;SETUP TESTS
                ;*****************************************************************

1221            ;*****************************************************************
                ;TEST 1  REFERENCE EACH REGISTER
                ;               REFERENCE EACH REGISTER BY A MOVE INSTRUCTION
                ;*****************************************************************

1265            ;*****************************************************************
                ;TEST 2  RHCS2-CONTROL AND STATUS 2

1268            ;       THIS PARTIALLY TESTS RHCS2 TO ENABLE DETERMINATION
                ;       OF THE NUMBER OF DRIVES PRESENT

1271            ;*****************************************************************

1290            ;*****************************************************************
                ;TEST 3  PARTIAL TEST FOR RHAS FOR UNIT NUMBERS PRESENT
                ;*****************************************************************

1308            ;*****************************************************************
                ;TEST 4  TEST FOR DRIVES PRESENT USING RHAS AND RHCS2
                ;*****************************************************************

1448            ;*****************************************************************
                ;TEST 5  TEST SERIAL NUMBER AND DRIVE TYPEI
                ;               READ SERIAL NUMBER REGISTER AND DRIVE TYPE REGISTER
                ;               TYPE IT OUT AND PROCEED
                ;               TO LOOP HERE SET SWITCH 8 AND THIS TEST NO AND RESTART

1454            ;*****************************************************************

1493            ;*****************************************************************
                ;TEST 6  CHECK MOL TO BE LOW

1496            ;       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

1500            ;*****************************************************************
```

```
                ;*****************************************************************
1525            ;REGISTER TESTS
                ;*****************************************************************

1527            ;*****************************************************************
                ;TEST 7  RHCS2 - CONTROL AND STATUS 2
                ;               TEST LOADING AND READING OF ALL POSSIBLE BITS IN THE HARDWARE
                ;               REGISTERS.  USE A PETTERN OF WALKING 0'S (-2,-3,-5 ETC.) AND
                ;               WALKING 1'S (1,2,4,10 ETC)
                ;*****************************************************************

1552            ;*****************************************************************
                ;TEST 10 RHCS1 - CONTROL AND STATUS 1 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)
                ;*****************************************************************

1574            ;*****************************************************************
                ;TEST 11 RHCS1 - BIT # 13 - MCPE
                ;       THIS FORCES A MASS BUS CONTROL PARITY ERROR
                ;       BY SETTING 'PAT', LOOKING FOR 'PAR', WRITING RHCS1
                ;       AND READING RHER1
                ;*****************************************************************

1628            ;*****************************************************************
                ;TEST 12 RHWC - WORD COUNT 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)
                ;*****************************************************************

1652            ;*****************************************************************
                ;TEST 13 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)
                ;*****************************************************************

1676            ;*****************************************************************
                ;TEST 14 RHER1 - ERROR REGISTER #1
                ;               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)
                ;*****************************************************************

1699            ;*****************************************************************
                ;TEST 15 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)
                ;*****************************************************************
```

```
1757    ;*****************************************************************
        TEST 16 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)
        ;*****************************************************************

1783    ;*****************************************************************
        TEST 17 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)
        ;*****************************************************************

1809    ;*****************************************************************
        TEST 20 RHOF - MARGIN/OFFSET 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)
        ;*****************************************************************

1834    ;*****************************************************************
        TEST 21 RHCA - DESIRED CYLINDER 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)
        ;*****************************************************************

1855    ;*****************************************************************
        TEST 22 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)
        ;*****************************************************************

1879    ;*****************************************************************
        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
        ;*****************************************************************

1889    ;*****************************************************************
        TEST 23 CONTROL AND STATUS 2 (RHCS 2) - 'NED'
                THIS TESTS THE UNIT SELECT BITS #0-2 (US1-4)
                AND NON-EXISTENT DRIVE BIT #12 (NED)

1894            THE OTHER RHCS2 BITS ARE NOT TESTED HERE
```

```
1896    ;*****************************************************************
        ;*****************************************************************
        ;*****************************************************************

2038    ;*****************************************************************
        TEST 24 CONTROL AND STATUS 2 (RHCS2) - 'CLR'

2048    ;*****************************************************************

2289    ;*****************************************************************
        TEST 25 PACK ACKNOWLEDGE COMMAND TEST

2292            THE PACK ACKNOWLEDGE 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

2297    ;*****************************************************************

2358    ;*****************************************************************
        TEST 26 UNIBUS INIT TEST

2361            ALL POSSIBLE REGISTER BITS ARE FILLED WITH ONES
                A RESET COMMAND IS GIVEN
                ALL REGISTERS ARE CHECKED

2365    ;*****************************************************************

        ;*****************************************************************
2450    SILO TESTS
        ;*****************************************************************

2453    ;*****************************************************************
        TEST 27 SILO TST 1

2456            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

2462    ;*****************************************************************

2506    ;*****************************************************************
        TEST 30 SILO TEST 2

2509            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

2519    ****************************************************************

2569    ****************************************************************
        TEST 31 SILO TEST 3

2572                    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.

2576    ****************************************************************

2636    ****************************************************************
        TEST 32 SILO TEST4
                        NOW PUT 67 WORDS INTO SILO AND CHECK FOR DLT
                        EVEN AFTER THE 67TH. WORD INPUT THE FIRST WORD SHOULD NOT CHANGE

2641    ****************************************************************

2667    ********************************  ********************************
        TEST 33 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 ARE DONE, ON THE LAST, 'DTL' IN RHCS2 SHOULD BE SET.

2674    ****************************************************************

        ****************************************************************
2742    MORE REGISTER TESTS
        ****************************************************************

2745    ****************************************************************
        TEST 34 TEST ODD BYTE INSTRUCTION ON RHCS1

2748                    RDY (BIT 07) AND DVA (BIT 11) SHOULD ALWAYS BE SET

2750    ****************************************************************

2780    ****************************************************************
        TEST 35 TEST ODD BYTE INSTRUCTION ON RHCS2

2783                    IR (BIT 06) AND THE UNIT SELECT (BIT 0-2) WILL BE SET

2785    ****************************************************************

2824    ****************************************************************
        TEST 36 ODD BYTE TEST ON RHWC
```

```
                2827            IN THIS REGISTER NO BITS SHOULD BE PERMANENTLY SET

                2829    ;*******************************************************************

                2857    ;*******************************************************************
                        TEST 37 TEST ODD BYTE INSTRUCTION ON RHBA

                2860            BIT 0 SHOULD ALWAYS BE 0

                2862    ;*******************************************************************

                        ;*******************************************************************
        2889    DCL COMMAND TESTS
                ;*******************************************************************

                2891    ;*******************************************************************
                                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

                2898            WHENEVER ANY OTHER USE IS MADE OF THESE REGISTERS
                                APPROPIATE SAVING MUST BE DONE

                2901    ;*******************************************************************

                2908    ;*******************************************************************
                                ERROR REGISTER #01 (RHER1) TEST
                                BIT #1 (ILLEGAL REGISTER) CANNOT BE TESTED ON PDP11 THIS BIT
                                IS FOR PDP10 USE ONLY
                        ;*******************************************************************

                2916    ;*******************************************************************
                        TEST 40 TEST ILF BIT #0 IN REG. RHER1

                2919            ALL 3 ILLEGAL FUNCTION CODES SHOULD SET - ATA,ERR,ILF - AND ARE TESTED
                                A GO WITHOUT CLEARING ILF ERR SHOULD SET - MXF,DLT,TRE - BITS AND THEY ARE ALSO

                2922    ;*******************************************************************

                3012    ;*******************************************************************
                        TEST 41 READ IN PRESET

                3015            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
```

```
3020    ;*****************************************************************

3090    ;*****************************************************************
        TEST 42 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

3097    ;*****************************************************************

3225    ;*****************************************************************
        TEST 43 DRIVE CLEAR

3228            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

3234    ;*****************************************************************

3502    ;*****************************************************************
        TEST 44 SEEK COMMAND TEST

3505            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

3510    ;*****************************************************************

3745    ;*****************************************************************
        TEST 45 UNLOAD COMMAND TEST

3748            THE UNLOAD COMMAND WILL BE LOADED INTO RHCS1 WITH GO
                THEN ALL REGISTERS WILL BE CHECKED
                RH CLEAR WILL BE GIVEN

3752    ;*****************************************************************

3875    ;*****************************************************************
        TEST 46 OFFSET COMMAND TEST

3878            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
```

```
3883    ;*****************************************************************
4018    ;*****************************************************************
        ;TEST 47 RETURN TO CENTER LINE COMMAND TEST
4021            ;THE RETURN TO CENTER LINE 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
4026    ;*****************************************************************
4149    ;*****************************************************************
        ;TEST 50 RECALIBRATE COMMAND TEST
4152            ;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
4157    ;*****************************************************************
4279    ;*****************************************************************
        ;TEST 51 RELEASE COMMAND TEST
4282            ;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
4287    ;*****************************************************************
4350    ;*****************************************************************
        ;TEST 52 MAKE CURRENT CYLINDER = 0
        ;*****************************************************************
4364    ;*****************************************************************
        ;TEST 53 LOOK AHEAD REGISTER
4367            ;A SEARCH COMMAND IS GIVEN FOR CYLINDER 0, TRACK 0, SECTOR 21.
                ;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 AFTER 256 MORE BYTES
                ;THE SECTOR COUNT FIELD IS CHECKED AFTER EACH SECTOR
                ;AT THE END ALL REGISTERS ARE CHECKED
4374    ;*****************************************************************
4577    ;*****************************************************************
        ;TEST 54 MAKE CURRENT CYLINDER = 0
        ;*****************************************************************
```

```
                ;*****************************************************************
4592            ;READ/WRITE ADDRESSING VIA RHMR
                ;*****************************************************************

4595            ;*****************************************************************
                ;TEST 55 WRITE HEADER AND DATA 1

4598            ;       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 PASS
                ;       BECAUSE ERROR LOGIC HAS NOT BEEN CHECKED YET (ON FIRST PASS)

4604            ;*****************************************************************

4706            ;*****************************************************************
                ;TEST 56 WRITE HEADER AND DATA 2

4709            ;       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)

4715            ;*****************************************************************

4827            ;*****************************************************************
                ;TEST 57 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)

4835            ;*****************************************************************

4951            ;*****************************************************************
                ;TEST 60 PROGRAM ERROR RHCS2 #10

4954            ;       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

4959            ;*****************************************************************

5020            ;*****************************************************************
                ;       THESE TESTS ARE THROUGH THE MAINTAINABILITY REGISTER - RHMR
```

```
                5023            THE SECTOR GAP AND SYNC BYTE ARE ALWAYS READ AS
                                ZEROS AND 144000 NO MATTER WHAT IS IN THE SIMULATED DISK AREA
                                TAGGED SECGAP: AND WSSYNC:

                5027            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

                5032            CRC IS READ FROM SIMULATED DISK LOCATION WCRC:
                                HEADER GAP IS ALWAYS READ AS ZEROS NO MATTER
                                WHAT IS ON THE SIMULATED DISK AREA

                5036            THE DATA SYNC IS READ  FROM HDWSYN:
                                ON SIMULATED DISK

                5039            ALL DATA IS READ FROM SIMULATED DISK DISK:
                        ;*******************************************************************

                5045    ;*******************************************************************
                        TEST 61 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)

                5052    ;*******************************************************************

                5172    ;*******************************************************************
                        TEST 62 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)

                5179    ;*******************************************************************

                5298    ;*******************************************************************
                        TEST 63 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)

                5305    ;*******************************************************************

                5424    ;*******************************************************************
                        TEST 64 WRITE DATA
```

```
5427            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

5432    ;*****************************************************************

5520    ;*****************************************************************
        TEST 65 READ DATA

5523            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)

5528    ;*****************************************************************

5635    ;*****************************************************************
        TEST 66 WRITE CHECK HEADER AND DATA

5638            WRITE CHECK CYLINDER 0, FORMAT 16 BITS PER WORD
                TRACK 1, SECTOR 1, KEYS 0, 36 WORDS AS SHOWN BELOW
                ANY DEVICE LOGIC ERROR INDICATION IS NOT CONCLUSIVE ON FIRST PASS
                BECAUSE ERROR LOGIC HAS NOT BEEN CHECKED YET (ON FIRST PASS)
                ONLY RH WRITE CHECK ERROR (RHCS2 BIT 14) IS TESTED HERE
        ;*****************************************************************

5766    ;*****************************************************************
        TEST 67 WRITE CHECK DATA

5769            WRITE CHECK DATA CYLINDER 0 FORMAT 16 BITS PER WORD
                TRACK 1, SECTOR 1, KEYS 0, 32 WORDS OF DATA
                ANY DEVICE LOGIC ERROR INDICATIONS ARE NOT CONCLUSIVE ON FIRST PASS
                BECAUSE ERROR LOGIC HAS NOT YET (ON FIRST PASS) BEEN CHECKED
                ONLY RH WRITE CHECK ERROR IS TESTED

5775    ;*****************************************************************

        ;*****************************************************************
5896    ERROR BIT FUNCTIONAL TESTS
        ;*****************************************************************

5900    ;*****************************************************************
        TEST 70 ATTENTION WITH ERROR TEST

5903            THIS TESTS THE SETTING OF ATA BIT BOTH IN THE RHAS
                AND THE RHDSI REGISTERS WITH THE SETTING OF EACH
                ERROR BIT ON THE THREE ERROR REGISTERS.
                IN EACH OF THE ABOVE CASES ERR IN RHDSI 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

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

;*****************************************************************
TEST 71 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)

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

;*****************************************************************
TEST 72 RHCS2 - BIT # '' - NEM

                READ CYLINDER0, FORMAT 16 BITS PER WORD
                TRACK0, SECTOR 1, KEYS 0, 1 WORD OF 177400
                THIS IS DONE WITH BUS ADDRESS INHIBIT SET
                BUS ADDRESS USED IS 760000 THIS IS ALWAYS NON EXISTANT
                THIS SHOULD SET NEM

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

;*****************************************************************
TEST 73 WRITE CHECK ERROR

                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

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

;*****************************************************************
TEST 74 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
```

```
6395    ************************************************************

6530    ************************************************************
        TEST 75 ERROR REGISTER #1-BIT 4 -FORMAT ERROR

6533            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.

6538    ************************************************************

6630    ************************************************************
        TEST 76 RHER1 - BIT #2 - REG. MODIFICATION REFUSED

6633            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

6647    ************************************************************

6733    ************************************************************
        TEST 77 MAKE CURRENT CYLINDER = 1
        ************************************************************

6747    ************************************************************
        TEST 100        ERROR REG1 - BIT #7 - HEADER COMPARE ERROR

6750            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

6759    ************************************************************
```

```
6813    ****************************************************************
        TEST 101        MAKE CURRENT CYLINDER = 0
        ****************************************************************

6827    ****************************************************************
        TEST 102        ERROR REG1 - BIT #7 - HEADER COMPARE ERROR

6830            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

6839    ****************************************************************

6893    ****************************************************************
        TEST 103        MAKE CURRENT CYLINDER = 1
        ****************************************************************

6907    ****************************************************************
        TEST 104        ERROR REG.1 - BIT #7 - HEADER COMPARE ERROR

6910            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

6919    ****************************************************************

6968    ****************************************************************
        TEST 105        MAKE CURRENT CYLINDER = 0
        ****************************************************************

6982    ****************************************************************
        TEST 106        ERROR REG.1 - BIT #7 - HEADER COMPARE ERROR

6985            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
```

```
6994    ;*****************************************************************

7042    ;*****************************************************************
        TEST 107        RHER1 - BIT #8 - CRC ERROR (READING)

7045            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

7054    ;*****************************************************************

7110    ;*****************************************************************
        TEST 110        RHER1 - BIT 8 - CRC ERROR (WRITING)

7113            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

7122    ;*****************************************************************

7171    ;*****************************************************************
        TEST 111        MAKE CURRENT CYLINDER = 410.
        ;*****************************************************************

7186    ;*****************************************************************
        TEST 112        RHDS1 (BIT #10) - LAST SECTOR TRANSFERRED, 'LST'
                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.

7194    ;*****************************************************************

7297    ;*****************************************************************
        TEST 113        ERROR REGISTER 1 - BIT #9 AOE

7300            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

7310    ;*****************************************************************

7437    ;*****************************************************************
        TEST 114        MAKE CURRENT CYLINDER = 0
        ;*****************************************************************

7451    ;*****************************************************************
        TEST 115        ERROR REGISTER 1 - BIT #10 IAE

7454            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

7460    ;*****************************************************************

7554    ;*****************************************************************
        TEST 116        ERROR REGISTER 1- BIT #10 IAE

7557            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 RHL  TO 0

                IAE BIT SHOULD SET

7566    ;*****************************************************************

7665    ;*****************************************************************
        TEST 117        ERROR REGISTER 1- BIT #10 IAE

7668            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

7677    ;*****************************************************************

7768    ;*****************************************************************
        TEST 120        END OF DRIVE
```

```
                7771            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

                7776    ;;;
```

```
                ;*****************************************************************
7930            ;SAVE REGISTERS ROUTINE
                ;*****************************************************************

                ;*****************************************************************
7960            ;FLOAT 1 AND 0
                ;*****************************************************************

                ;*****************************************************************
8019            ;CLEAR MEMORY ROUTINE
                ;*****************************************************************

                ;*****************************************************************
8052            ;LOCAL TRAPS
                ;*****************************************************************

                ;*****************************************************************
8069            ;CLEAR DISK ROUTINE
                ;*****************************************************************

                ;*****************************************************************
8082            ;CHECK DISK STATUS ROUTINES
                ;*****************************************************************

                ;*****************************************************************
8212            ;SAVE ROUTINE
                ;*****************************************************************

                ;*****************************************************************
8240            ;WRITE CHECK ROUTINE
                ;*****************************************************************

                ;*****************************************************************
8284            ;COMPARE ROUTINE
                ;*****************************************************************

                ;*****************************************************************
8380            ;CRC GENERATION ROUTINE
                ;*****************************************************************
```

```
                ;*****************************************************************
8698            ;JAM CURRENT CYLINDER ROUTINE
                ;*****************************************************************

                ;*****************************************************************
8737            ;ECC GENERATION AND COMPARISON ROUTINE
                ;*****************************************************************

                ;*****************************************************************
9077            ;RH BASE ADDRESS CHANGE ROUTINE
                ;*****************************************************************

9079                    ;THIS ROUTINE WILL ALLOW THE CHANGE OF THE BASE
                        ;ADDRESS FROM 176700 TO ANY TYPED VALUE

                ;*****************************************************************
9156            ;DISK SIMULATION
                ;*****************************************************************

9157            ;*****************************************************************
                ;*****************************************************************

9217            ;*****************************************************************
                ;*****************************************************************

9314            ;*****************************************************************

9336            ;*****************************************************************

9343            ;*****************************************************************
                ;*****************************************************************

9470            ;*****************************************************************
                ;*****************************************************************

9537            ;*****************************************************************
                ;*****************************************************************

9608            ;*****************************************************************
                ;*****************************************************************

9686            ;*****************************************************************
                ;*****************************************************************

9718            ;*****************************************************************
                ;*****************************************************************

9850            ;*****************************************************************
                ;*****************************************************************
```

```
                10010   ;*****************************************************************
                        ;*****************************************************************

                10123   ;*****************************************************************
                        ;*****************************************************************

                        ;*****************************************************************
        10168   ;SYSMAC LIBRARY ROUTINES
                        ;*****************************************************************

                10170   ;*****************************************************************

                        ;*****************************************************************
        10172   ;SCOPE HANDLER ROUTINE
                        ;*****************************************************************

                10174   ;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

                10244   ;*****************************************************************

                        ;*****************************************************************
        10246   ;CONVERT BINARY TO DECIMAL AND TYPE ROUTINE
                        ;*****************************************************************

                10248   ;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

                10312   ;*****************************************************************
```

```
                ;*****************************************************************
10314           ;TYPE ROUTINE
                ;*****************************************************************

10316           ;ROUTINE TO TYPE ASCIZ MESSAGE. MESSAGE MUST TERMINATE WITH A 0 BYTE.
                ;THE ROUTINE WILL INSERT A NUMBER OF NULL CHARACTERS AFTER A LINE FEED.
                ;NOTE1:  $NULL CONTAINS THE CHARACTER TO BE USED AS THE FILLER CHARACTER.
                ;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 ASCIZ STRING
                ;OR
                ;       TYPE
                ;       MESADR

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

10359           ;*****************************************************************

                ;*****************************************************************
10361           ;TTY INPUT ROUTINE
                ;*****************************************************************

10370           ;TK INITIALIZE ROUTINE
                ;THIS ROUTINE WILL INITIALIZE THE TTY KEYBOARD INPUT QUEUE
                ;SETUP THE INTERRUPT VECTOR AND TURN ON THE KEYBOARD INTERRUPT
                ;CALL
                ;       JSR     PC,$TKINT
                ;       RETURN

10387           ;TK SERVICE ROUTINE
                ;THIS ROUTINE WILL SERVICE THE TTY KEYBOARD INTERRUPT

10409           ;*****************************************************************
                ;THIS ROUTINE WILL INPUT A SINGLE CHARACTER FROM THE TTY
                ;CALL:
                ;       RDCHR                   ;;INPUT A SINGLE CHARACTER FROM THE TTY

10413           ;       RETURN HERE             ;;CHARACTER IS ON THE STACK

10429           ;*****************************************************************
                ;THIS ROUTINE WILL INPUT A STRING FROM THE TTY
                ;CALL:
                ;       RDLIN                   ;;INPUT A STRING FROM THE TTY
                ;       RETURN HERE             ;;ADDRESS OF FIRST CHARACTER WILL BE ON THE STACK
                ;                               ;;TERMINATOR WILL BE A BYTE OF ALL 0'S
```

```
                10462   ;*****************************************************************

                        ;*****************************************************************
        10464   ;READ AN OCTAL NUMBER FROM THE TTY
                        ;*****************************************************************

                10466   ;THIS ROUTINE WILL READ AN OCTAL (ASCII) NUMBER FROM THE TTY AND
                        ;CHANGE IT TO BINARY.
                        ;THE INPUT CHARACTERS WILL BE CHECKED TO INSURED THEY ARE LEGAL
                        ;OCTAL DIGITS. IF AN ILLEGAL CHARACTER IS READ A "?" WILL BE TYPED
                        ;FOLLOWED BY A CARRIAGE RETURN-LINE FEED. THE COMPLETE NUMBER MUST
                        ;THEN BE RETYPED. THE INPUT IS TERMINATED BY TYPING A CARRIAGE RETURN.
                        ;CALL:
                        ;       RDOCT                   ;;READ AN OCTAL NUMBER
                        ;       RETURN HERE             ;;LOW ORDER BITS ARE ON TOP OF THE STACK
                        ;                               ;;HIGH ORDER BITS ARE IN $HIOCT

                10516   ;*****************************************************************

                        ;*****************************************************************
        10518   ;ERROR HANDLER ROUTINE
                        ;*****************************************************************

                10520   ;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       ;;E       AND N=ERROR ITEM NUMBER

                10559   ;*****************************************************************

                        ;*****************************************************************
        10561   ;ERROR MESSAGE TYPEOUT ROUTINE
                        ;*****************************************************************

                10563   ;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.

                10616   ;*****************************************************************
```

```
                ;*****************************************************************
10618           ;BINARY TO OCTAL (ASCII) AND TYPE
                ;*****************************************************************

10620           ;THIS ROUTINE IS USED TO CHANGE A 16-BIT BINARY NUMBER TO A 6-DIGIT
                ;OCTAL (ASCII) NUMBER AND TYPE IT.
                ;$TYPOS---ENTER HERE TO SETUP SUPPRESS ZEROS AND NUMBER OF DIGITS TO TYPE
                ;CALL:
                ;       MOV     NUM,-(SP)       ;;NUMBER TO BE TYPED
                ;       TYPOS                   ;;CALL FOR TYPEOUT
                ;       .BYTE   N               ;;N=1 TO 6 FOR NUMBER OF DIGITS TO TYPE
                ;       .BYTE   M               ;;M=1 OR 0
                ;                               ;;1=TYPE LEADING ZEROS
                ;                               ;;0=SUPPRESS LEADING ZEROS

                ;$TYPON----ENTER HERE TO TYPE OUT WITH THE SAME PARAMETERS AS THE LAST
                ;$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
                ;       TYPOC                   ;;CALL FOR TYPEOUT

10694           ;*****************************************************************

                ;*****************************************************************
10696           ;TRAP DECODER
                ;*****************************************************************

10698           ;THIS ROUTINE WILL PICKUP THE LOWER BYTE OF THE "TRAP" INSTRUCTION
                ;AND USE IT TO INDEX THROUGH THE TRAP TABLE FOR THE STARTING ADDRESS
                ;OF THE DESIRED ROUTINE. THEN USING THE ADDRESS OBTAINED IT WILL
                ;GO TO THAT ROUTINE.

                ;*****************************************************************
10711           ;TRAP TABLE
                ;*****************************************************************

10713           ;THIS TABLE CONTAINS THE STARTING ADDRESSES OF THE ROUTINES CALLED
                ;BY THE "TRAP" INSTRUCTION.

10730           ;*****************************************************************
```

```
                ;*********************************************************************
10732           ;POWER DOWN AND UP ROUTINES
                ;*********************************************************************

10772           ;*********************************************************************
                ;*********************************************************************
```

```
    42
    43          ***ERROR TABLE, BIT DEFINITIONS & STARTING ADDRESSES***
    44
    47          OPERATIONAL SWITCH SETTINGS
    62          BASIC DEFINITIONS
   168          TRAP CATCHER
   175          STARTING ADDRESS(ES)
   188          MEMORY MANAGEMENT DEFINITIONS
   228          COMMON TAGS
   281          ERROR POINTER TABLE
   965          REGISTER ADDRESSES
  1123
  1124          ***DIAGNOSTIC CODE***
  1125
  1127          SETUP TESTS
  1222          T1      REFERENCE EACH REGISTER
  1267          T2      RHCS2-CONTROL AND STATUS 2
  1293          T3      PARTIAL TEST FOR RHAS FOR UNIT NUMBERS PRESENT
  1312          T4      TEST FOR DRIVES PRESENT USING RHAS AND RHCS2
  1453          T5      TEST SERIAL NUMBER AND DRIVE TYPEI
  1499          T6      CHECK MOL TO BE LOW
  1532          REGISTER TESTS
  1534          T7      RHCS2 - CONTROL AND STATUS 2
  1560          T10     RHCS1 - CONTROL AND STATUS 1 REGISTER
  1583          T11     RHCS1 - BIT # 13 - MCPE
  1638          T12     RHWC - WORD COUNT REGISTER
  1663          T13     RHBA - UNIBUS ADDRESS REGISTER
  1688          T14     RHER1 - ERROR REGISTER #1
  1712          T15     RHMR - MAINTENANCE REGISTER
  1771          T16     RHDST - DESIRED SECTOR/TRACK ADDRESS
  1798          T17     RHER2 - ERROR REGISTER #2
  1825          T20     RHOF - MARGIN/OFFSET REGISTER
  1851          T21     RHCA - DESIRED CYLINDER REGISTER
  1873          T22     RHER3 - ERROR REGISTER #3
  1908          T23     CONTROL AND STATUS 2 (RHCS 2) - 'NED'
  2058          T24     CONTROL AND STATUS 2 (RHCS2) - 'CLR'
  2310          T25     PACK ACKNOWLEDGE COMMAND TEST
  2380          T26     UNIBUS INIT TEST
  2473          SILO TESTS
  2476          T27     SILO TST 1
  2530          T30     SILO TEST 2
  2594          T31     SILO TEST 3
  2662          T32     SILO TEST4
  2694          T33     SILO TEST 5
  2770          MORE REGISTER TESTS
  2773          T34     TEST ODD BYTE INSTRUCTION ON RHCS1
  2809          T35     TEST ODD BYTE INSTRUCTION ON RHCS2
  2854          T36     ODD BYTE TEST ON RHWC
  2888          T37     TEST ODD BYTE INSTRUCTION ON RHBA
  2921          DCL COMMAND TESTS
  2948          T40     TEST ILF BIT #0 IN REG. RHER1
  3045          T41     READ IN PRESET
  3124          T42     NO OPERATION FUNCTION TEST
  3260          T43     DRIVE CLEAR
```

3538 T44 SEEK COMMAND TEST
3782 T45 UNLOAD COMMAND TEST
3913 T46 OFFSET COMMAND TEST
4057 T47 RETURN TO CENTER LINE COMMAND TEST
4189 T50 RECALIBRATE COMMAND TEST
4320 T51 RELEASE COMMAND TEST
4392 T52 MAKE CURRENT CYLINDER = 0
4407 T53 LOOK AHEAD REGISTER
4621 T54 MAKE CURRENT CYLINDER = 0
4637 READ/WRITE ADDRESSING VIA RHMR
4640 T55 WRITE HEADER AND DATA 1
4752 T56 WRITE HEADER AND DATA 2
4874 T57 WRITE HEADER AND DATA 3
4999 T60 PROGRAM ERROR RHCS2 #10
5094 T61 READ HEADER AND DATA 1
5222 T62 READ HEADER AND DATA 2
5349 T63 READ HEADER AND DATA 3
5476 T64 WRITE DATA
5573 T65 READ DATA
5689 T66 WRITE CHECK HEADER AND DATA
5821 T67 WRITE CHECK DATA
5952 ERROR BIT FUNCTIONAL TESTS
5956 T70 ATTENTION WITH ERROR TEST
6091 T71 BUS ADDRESS INHIBIT
6206 T72 RHCS2 - BIT # 11 - NEM
6299 T73 WRITE CHECK ERROR
6447 T74 ERROR REGISTER #1-BIT 4 -FORMAT ERROR
6591 T75 ERROR REGISTER #1-BIT 4 -FORMAT ERROR
6692 T76 RHER1 - BIT #2 - REG. MODIFICATION REFUSED
6796 T77 MAKE CURRENT CYLINDER = 1
6811 T100 ERROR REG1 - BIT #7 - HEADER COMPARE ERROR
6878 T101 MAKE CURRENT CYLINDER = 0
6893 T102 ERROR REG1 - BIT #7 - HEADER COMPARE ERROR
6960 T103 MAKE CURRENT CYLINDER = 1
6975 T104 ERROR REG.1 - BIT #7 - HEADER COMPARE ERROR
7037 T105 MAKE CURRENT CYLINDER = 0
7052 T106 ERROR REG.1 - BIT #7 - HEADER COMPARE ERROR
7113 T107 RHER1 - BIT #8 - CRC ERROR (READING)
7182 T110 RHER1 - BIT 8 - CRC ERROR (WRITING)
7244 T111 MAKE CURRENT CYLINDER = 410.
7260 T112 RHDS1 (BIT #10) - LAST SECTOR TRANSFERRED, 'LST'
7372 T113 ERROR REGISTER 1 - BIT #9 AOE
7513 T114 MAKE CURRENT CYLINDER = 0
7528 T115 ERROR REGISTER 1 - BIT #10 IAE
7632 T116 ERROR REGISTER 1- BIT #10 IAE
7744 T117 ERROR REGISTER 1- BIT #10 IAE
7848 T120 END OF DRIVE
7894
7895 ***SUBROUTINES***
7896
7901 END OF PASS ROUTINE
8011 SAVE REGISTERS ROUTINE
8041 FLOAT 1 AND 0
8100 CLEAR MEMORY ROUTINE

8133	LOCAL TRAPS
8150	CLEAR DISK ROUTINE
8163	CHECK DISK STATUS ROUTINES
8293	SAVE ROUTINE
8321	WRITE CHECK ROUTINE
8365	COMPARE ROUTINE
8461	CRC GENERATION ROUTINE
8779	JAM CURRENT CYLINDER ROUTINE
8818	ECC GENERATION AND COMPARISON ROUTINE
9158	RH BASE ADDRESS CHANGE ROUTINE
9237	DISK SIMULATION
10249	SYSMAC LIBRARY ROUTINES
10253	SCOPE HANDLER ROUTINE
10328	CONVERT BINARY TO DECIMAL AND TYPE ROUTINE
10396	TYPE ROUTINE
10443	TTY INPUT ROUTINE
10546	READ AN OCTAL NUMBER FROM THE TTY
10600	ERROR HANDLER ROUTINE
10643	ERROR MESSAGE TYPEOUT ROUTINE
10700	BINARY TO OCTAL (ASCII) AND TYPE
10778	TRAP DECODER
10793	TRAP TABLE
10814	POWER DOWN AND UP ROUTINES

```
                        .TITLE  MAINDEC-11-DZRPS-D, RJP04 DISKLESS RH11 TEST-PART 1
                        ;#COPYRIGHT (C) 1975
                        ;#DIGITAL EQUIPMENT CORP.
                        ;#MAYNARD, MASS. 01754
                        ;#
                        ;#PROGRAM BY SUB MALLICK, PETE BLACKSTONE
                        ;#
                        ;#THIS PROGRAM WAS ASSEMBLED USING THE PDP-11 MAINDEC SYSMAC
                        ;#PACKAGE (MAINDEC-11-DZQAC-A3).
                        ;#


    ;/#\:/#\:/#\:/#\:/#\:/#\:/#\:/#\:/#\:/#\:/#\:/#\:/#\:/#\:/#\:/#\:/#\:/#\:/#\:/#\:/#\:
    ;/#\:/#\:/#\:/#\:/#\:/#\:/#\:/#\:/#\:/#\:/#\:/#\:/#\:/#\:/#\:/#\:/#\:/#\:/#\:/#\:/#\:
    ;/#\:/#\:/#\:/#\:/#\:/#\:/#\:/#\:/#\:/#\:/#\:/#\:/#\:/#\:/#\:/#\:/#\:/#\:/#\:/#\:/#\:

    ;DRIVE MUST BE LOCKED ON PORT A OR PORT B

    ;/:\#/:\#/:\#/:\#/:\#/:\#/:\#/:\#/:\#/:\#/:\#/:\#/:\#/:\#/:\#/:\#/:\#/:\#/:\#/:\#/:\*
    ;/:\#/:\#/:\#/:\#/:\#/:\#/:\#/:\#/:\#/:\#/:\#/:\#/:\#/:\#/:\#/:\#/:\#/:\#/:\#/:\#/:\*
    ;/:\#/:\#/:\#/:\#/:\#/:\#/:\#/:\#/:\#/:\#/:\#/:\#/:\#/:\#/:\#/:\#/:\#/:\#/:\#/:\#/:\*



                                        ;INTERNAL PROGRAM MACROS BEGIN HERE
                                        ;**********************************



                        ;*****************************************************************
                        ;*
                        ;*NOTE: MACROS BEGINNING WITH ".$" ARE SUPPLIED BY AN
                        ;*      EXTERNAL SYSMAC.SML SYSTEM MACRO PACKAGE WHICH
                        ;*      MUST BE MADE AVAILABLE TO THE SOURCE PROGRAM
                        ;*      AT ASSEMBLY TIME.
                        ;*
                        ;*****************************************************************


                        .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
177776                          PS=     177776          ;;PROCESSOR STATUS WORD
                                .EQUIV  PS,PSW
177774                          STKLMT= 177774          ;;STACK LIMIT REGISTER
177772                          PIRQ=   177772          ;;PROGRAM INTERRUPT REQUEST REGISTER
177570                          SWR=    177570          ;;SWITCH REGISTER
177570                          DISPLAY=SWR

                                ;*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

                                .SBTTL  STARTING ADDRESS(ES)
        000200                          .=200

000200  000137  004220                  JMP     @#BEGIN         ;;JUMP TO STARTING ADDRESS OF PROGRAM
        000210                          .=210                           ;ADDRESSES
000210  000137  004210                  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
                                ;
                                ;*KERNAL "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
                                ;
                                ;*KERNAL "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
                                ;
                                ;*****************************************************************
001110                                  .=1110                          ; ?
                                ;
```

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

                                .SBTTL  COMMON TAGS

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

                    000046                  .=46
000046  037310                  $ENDAD                  ;;LOGICAL END OF 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
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  OF 'GOOD' DATA
001122  000000                  $BDADR: .WORD   0        ;;CONTAINS  OF 'BAD' DATA
001124  000000                  $GDDAT: .WORD   0        ;;CONTAINS 'GOOD' DATA
001126  000000                  $BDDAT: .WORD   0        ;;CONTAINS 'BAD' DATA
001130  000000  000000  000000          .WORD   0,0,0    ;;RESERVED--NOT TO BE USED
001136  177560                  $TKS:   177560           ;;TTY KBD STATUS
001140  177562                  $TKB:   177562           ;;TTY KBD BUFFER
001142  177564                  $TPS:   177564           ;;TTY PRINTER STATUS REG.
001144  177566                  $TPB:   177566           ;;TTY PRINTER BUFFER REG.
001146  000                     $NULL:  .BYTE   0        ;;CONTAINS NULL CHARACTER FOR FILLS
001147  002                     $FILLS: .BYTE   2        ;;CONTAINS # OF FILLER CHARACTERS REQUIRED
001150  012                     $FILLC: .BYTE   12       ;;INSERT FILL CHARS. AFTER A "LINE FEED"
001151  000                     $TPFLG: .BYTE   0        ;;"TERMINAL AVAILABLE" FLAG (BIT<07>=0=YES)
001152  000000                  $REGAD: .WORD   0        ;;CONTAINS THE  FROM
                                                         ;;WHICH  ($REG0) WAS OBTAINED
001154  000000                  $REG0:  .WORD   0        ;;CONTAINS (($REGAD)+0)
001156  000000                  $REG1:  .WORD   0        ;;CONTAINS (($REGAD)+2)
001160  000000                  $REG2:  .WORD   0        ;;CONTAINS (($REGAD)+4)
001162  000000                  $REG3:  .WORD   0        ;;CONTAINS (($REGAD)+6)
001164  000000                  $REG4:  .WORD   0        ;;CONTAINS (($REGAD)+10)
001166  000000                  $REG5:  .WORD   0        ;;CONTAINS (($REGAD)+12)
001170  000000                  $TMP0:  .WORD   0        ;;USER DEFINED
001172  000000                  $TMP1:  .WORD   0        ;;USER DEFINED
001174  000000                  $TMP2:  .WORD   0        ;;USER DEFINED
001176  000000                  $TMP3:  .WORD   0        ;;USER DEFINED
001200  000000                  $TMP4:  .WORD   0        ;;USER DEFINED
001202  000000                  $TMP5:  .WORD   0        ;;USER DEFINED
001204  000000                  $TIMES: 0                ;;MAX. NUMBER OF ITERATIONS
001206  000000                  $ESCAPE:0                ;;ESCAPE ON ERROR
001210  177607  000377            $BELL:  .ASCIZ  <207><377><377>  ;;CODE FOR BELL
001214  077                     $QUES:  .ASCII  /?/      ;;QUESTION MARK
```

```
   276  001215     015                SCRLF:  .ASCII  <15>          ;;CARRIAGE RETURN
   277  001216  000012                SLF:    .ASCIZ  <12>          ;;LINE FEED
```

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

                                .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

001220                          $ERRTB:


                                ;ITEM1
001220  054216                          EM1                     ;WRONG DATA IN READING OR WRITING HARDWARE REGISTER
001222  057461                          DH1                     ;PC
                                                                ;REG. ADDR.
                                                                ;GOOD DATA
                                                                ;RECEIVED DATA
001224  064040                          DT1                     ;$ERRPC,REGADR,$GDDAT,$BDDAT
001226  064556                          DF1                     ;0,0,0,0,0


                                ;ITEM2
001230  054301                          EM2                     ;ERROR ON  DATA COMMAND

001232  062615                          DH33                    ;PC
                                                                ;PC OF JSR
                                                                ;TEST NO
                                                                ;WORD NO.
                                                                ;GOOD DATA
                                                                ;CONTENTS OF RHCS1
                                                                ;CONTENTS OF RHDS1
                                                                ;CONTENTS OF RHER1
001234  064420                          DT33                    ;$ERRPC,PCJSR,$TSTNM,ERWORD,$GDDAT,CS1,DS1,ER1
001236  064726                          DF33                    ;0,0,0,1,0,0,0,0


                                ;ITEM3
001240  054301                          EM2                     ;ERROR ON  DATA COMMAND

001242  062372                          DH32                    ;PC
                                                                ;PC OF JSR
                                                                ;TEST NO
```

```
                                                                ;WORD NO.
                                                                ;GOOD DATA
                                                                ;BAD DATA
                                                                ;CONTENTS OF RHCS1
                                                                ;CONTENTS OF RHDS1
                                                                ;CONTENTS OF RHER1

001244  064374                          DT32                  ;$ERRPC,PCJSR,$TSTNM,ERWORD,$GDDAT,$BDDAT,CS1,DS1,ER1
001246  064715                          DF32                  ;0,0,0,1,0,0,0,0,0,

                              ;ITEM4
001250  054301                          EM2                   ;ERROR ON  DATA COMMAND

001252  062170                          DH31                  ;PC
                                                              ;TEST NO
                                                              ;WORD NO.
                                                              ;GOOD DATA
                                                              ;BAD DATA
                                                              ;CONTENTS OF RHCS1
                                                              ;CONTENTS OF RHDS1
                                                              ;CONTENTS OF RHER1

001254  064352                          DT31                  ;$ERRPC,$TSTNM,ERWORD,$GDDAT,$BDDAT,CS1,DS1,ER1
001256  064705                          DF31                  ;0,0,1,0,0,0,0,0,


                              ;ITEM5
001260  000000                          0                     ;
001262  000000                          0                     ;
001264  064352                          DT31                  ;$ERRPC,$TSTNM,ERWORD,$GDDAT,$BDDAT,CS1,DS1,ER1
001266  064705                          DF31                  ;0,0,1,0,0,0,0,0,


                              ;ITEM6
001270  054330                          EM6                   ;ERROR ON WRITE HEADER AND DATA

001272  062372                          DH32                  ;PC
                                                              ;PC OF JSR
                                                              ;TEST NO
                                                              ;WORD NO.
                                                              ;GOOD DATA
                                                              ;BAD DATA
                                                              ;CONTENTS OF RHCS1
                                                              ;CONTENTS OF RHDS1
                                                              ;CONTENTS OF RHER1

001274  064374                          DT32                  ;$ERRPC,PCJSR,$TSTNM,ERWORD,$GDDAT,$BDDAT,CS1,DS1,ER1
001276  064715                          DF32                  ;0,0,0,1,0,0,0,0,0,


                              ;ITEM7
```

```
001300  054330                          EM6             ;ERROR ON WRITE HEADER AND DATA
001302  057604                          DH2             ;PC
                                                        ;TEST NO
                                                        ;WORD NO.
                                                        ;GOOD DATA
                                                        ;BAD DATA
001304  064066                          DT3             ;$ERRPC,$TSTNM,ERWORD,$GDDAT,$BDDAT
001306  064567                          DF3             ;0,0,1,0,0,

                                ;ITEM10
001310  000000                          0               ;
001312  000000                          0               ;
001314  064066                          DT3             ;$ERRPC,$TSTNM,ERWORD,$GDDAT,$BDDAT
001316  064567                          DF3             ;0,0,1,0,0,

                                ;ITEM11
001320  054367                          EM11            ;CONTROLLER OR DRIVE STATUS
001322  057727                          DH11            ;PC
                                                        ;TEST NO
                                                        ;FAILING REG. ADDR
                                                        ;CONTENTS OF RHCS1
                                                        ;CONTENTS OF RHCS2
                                                        ;CONTENTS OF RHDS1
                                                        ;CONTENTS OF RHER1
001324  064102                          DT11            ;$ERRPC,$TSTNM,$BDADR,CS1,CS2,DS1,ER1
001326  064574                          DF11            ;0,0,0,0,0,0

                                ;ITEM12
001330  054367                          EM11            ;WRONG DATA FROM SILO

001332  057461                          DH1             ;PC
                                                        ;REG.ADDR
                                                        ;GOOD DATA
                                                        ;RECEIVED DATA
001334  064040                          DT1             ;$ERRPC,REGADR,$GDDAT,$BDDAT
001336  064556                          DF1             ;0,0,0,0

                                ;ITEM13
001340  000000                          0
001342  000000                          0
001344  064040                          DT1             ;$ERRPC,TSTNM,REGADR,$GDDAT,$BDDAT
001346  064556                          DF1             ;0,0,0,0,0

                                ;ITEM14
001350  054422                          EM14            ;REGISTER FAILED
001352  060107                          DH14            ;PC
                                                        ;FAILING REG. ADDR
                                                        ;CONTENTS OF FAILING REG.
                                                        ;CONTENTS OF RHCS1
```

```
                                                        ;CONTENTS OF RHCS2
                                                        ;CONTENTS OF RHDS1
                                                        ;CONTENTS OF RHER1
001354  064122                      DT14              ;$ERRPC,$BDADR,$BDDAT,CS1,CS2,DS1,ER1
001356  064603                      DF14              ;0,0,0,0,0,0,0


                            ;ITEM15
001360  054442                      EM15              ;SPECIFIED REG. NON EXISTANT SO ABORT
                                                      ;PROGRAM
001362  060307                      DH15              ;PC
                                                      ;ADDR. OF REG
001364  064144                      DT15              ;$ERRPC,TEMP1
001366  064613                      DF15              ;0,0


                            ;ITEM16
001370  054513                      EM16              ;WAIT LOOP FAILED
001372  060340                      DH16              ;PC
                                                      ;WAT PC
                                                      ;BIT WANTED
                                                      ;REG. ADR.
                                                      ;REG. CONT.
001374  064154                      DT16              ;$ERRPC,$TMP3,$TMP1,$TMP0,$BDDAT
001376  064616                      DF16              ;0,0,0,0


                            ;ITEM17
001400  054534                      EM17              ;WRITE CHECK FAILING
001402  060503                      DH17              ;PC
                                                      ;TEST NO
                                                      ;CONTENTS OF RHBA
                                                      ;CONTENTS OF RHDB
                                                      ;CONTENTS OF RHWC
                                                      ;CONTENTS OF RHCS1
                                                      ;CONTENTS OF RHCS2
001404  064172                      DT17              ;$ERRPC,$TSTNM,$BA,DB,WC,CS1,CS2
001406  064623                      DF17              ;0,0,0,0,0,0,0


                            ;ITEM20
001410  054560                      EM20              ;REGISTER FAILING
001412  060666                      DH20              ;PC
                                                      ;TST NO
                                                      ;CONTENTS OF RHER1
                                                      ;CONTENTS OF RHER2
                                                      ;CONTENTS OF RHER3
                                                      ;CONTENTS OF RHAS
                                                      ;CONTENTS OF RHDS1
001414  064212                      DT20              ;$ERRPC,TSTNM ER1,ER2,ER3,AS,DS1
001416  064632                      DF20              ;0,0,0,0,0,0,0


                            ;ITEM21
```

```
001420  054601                          EM21            ;INTERRUPT FAILING
001422  061051                          DH21            ;PC
                                                        ;TEST NO
                                                        ;CONTENTS OF RHCS1
                                                        ;CONTENTS OF RHAS
                                                        ;CONTENTS OF RHDS1
001424  064232                          DT21            ;$ERRPC,TSTNM,CS1,AS,DS1
001426  064641                          DF21            ;0,0,0,0,0


                        ;ITEM22
001430  054623                          EM22            ;ERROR IN DRIVE PRESENT -
                                                        ;LOOKING AT RHAS AND RHCS2-NED(BIT#12)
                                                        ;DRIVES PRESENT DO NOT AGREE
                                                        ;NOTE: ON DUAL PORT SYSTEM
                                                        ;DRIVE ON OTHER PORT WILL NOT GIVE NED
                                                        ;HENCE THERE WILL BE A MISSMATCH
001432  061171                          DH22            ;PC
                                                        ;TEST NO
                                                        ;RHAS UNIT (RHER1 BITS SET)
                                                        ;RHCS2 UNIT ('NED' BIT TEST)
001434  064246                          DT22            ;$ERRPC,TSTNM
001436  064646                          DF22            ;0,0


                        ;ITEM23
001440  000000                          0               ;NO LONGER USED DUE TO SPECIAL 'NED'
001442  000000                          0               ;TEST TABLE TYPE OUT ROUTINE
001444  000000                          0
001446  000000                          0


                        ;ITEM 24
001450  055272                          EM24            ;LOOK AHEAD REGISTER AT THE
                                                        ;BEGINNING OF A SECTOR IS IN
                                                        ;ERROR
001452  061274                          DH24            ;PC
                                                        ;RHDST
                                                        ;BAD RHLA
                                                        ;GOOD RHLA
                                                        ;SECTOR NO
                                                        ;SECTOR CLOCK
001454  064254                          DT24            ;$ERRPC,DST,$BDDAT,$TMP1,$TMP2,$TMP3
001456  064652                          DF24            ;0,0,0,0,0


                        ;ITEM 25
001460  055365                          EM25            ;LOOK AHEAD REGISTER IS
                                                        ;IN ERROR


001462  061274                          DH24            ;PC
                                                        ;RHDST
                                                        ;BAD RHLA
                                                        ;GOOD RHLA
                                                        ;SECTOR NO
```

```
                                                        ;SECTOR CLOCK
001464  064254                          DT24            ;$ERRPC,DST,$BDDAT,$TMP1,$TMP2,$TMP3
001466  064652                          DF24            ;0,0,0,0,0
                                ;ITEM26
001470  054367                          EM11            ;CONTROLLER OR DRIVE STATUS

001472  061457                          DH26            ;PC
                                                        ;PC OF JSR
                                                        ;FAILING REGISTER ADDRESS
                                                        ;CONTENTS OF RHCS1
                                                        ;CONTENTS OF RHCS2
                                                        ;CONTENTS OF RHDS1
                                                        ;CONTENTS OF RHER1

001474  064274                          DT26            ;$ERRPC,PCJSR,$BDADR,CS1,CS2,DS1,ER1
001476  064661                          DF26            ;0,0,0,0,0,0,

                                ;ITEM27
001500  054216                          EM1             ;ERROR IN READING OR WRITING HARDWARE REGISTER

001502  061662                          DH27            ;PC
                                                        ;PC OF JSR
                                                        ;TEST NUMBER
                                                        ;FAILING REGISTER
                                                        ;GOOD DATA
                                                        ;RECEIVED DATA

001504  064316                          DT27            ;$ERRPC,PCJSR,TSTNM,REGADR,$GDDAT,$BDDAT
001506  064671                          DF27            ;0,0,0,0,0,0

                                ;ITEM30
001510  055425                          EM30            ;CURRENT CYLINDER DOES NOT REFLECT DESIRED CYLINDER REG.
001512  062025                          DH30            ;PC
                                                        ;PC OF JSR
                                                        ;REGISTER ADDRESS
                                                        ;GOOD DATA
                                                        ;BAD DATA

001514  064334                          DT30            ;$ERRPC,PCJSR,REGADR,$GDDAT,$BDDAT
001516  064677                          DF30            ;0,0,0,0,0

                                ;ITEM31
001520  055547                          EM31            ;ECC GENERATED IS INCORRECT
                                                        ;EVERY WORD IN THIS SECTOR IS GIVEN IN "DATA USED"

001522  063020                          DH34            ;PC
                                                        ;TEST NUMBER
                                                        ;GOOD ECC1
```

```
                                                        ;GOOD EC2C
                                                        ;WRITTEN ECC1
                                                        ;WRITTEN ECC2
                                                        ;DATA USED

001524  064442                          DT34            ;SERRPC,TSTNM,GECC1,GECC2,WECC1,WECC2,DISK

001526  064736                          DF34            ;0,0,0,0,0,0,0


                                ;ITEM32
001530  055672                          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

001532  063203                          DH35            ;PC
                                                        ;TEST NUMBER
                                                        ;GOOD ECC1
                                                        ;GOOD ECC2
                                                        ;PATTERN REGISTER
                                                        ;RHER1

001534  064462                          DT35            ;SERRPC,TSTNM,GECC1,GECC2,EC2 ER1

001536  064745                          DF35            ;0,0,0,0,0,0


                                ;ITEM33
001540  056161                          EM33            ;HIGH COUNT BIT NOT HIGH AFTER 38859 CLOCKS
001542  063406                          DH36            ;PC
                                                        ;PC OF JSR
                                                        ;TEST NUMBER
                                                        ;RHMR
                                                        ;POSITION REG.
                                                        ;PATTERN REGISTER

001544  064504                          DT36            ;SERRPC,PCJSR,TSTNM,MR,EC1,EC2

001546  064755                          DF36            ;0,0,0,0,0,0


                                ;ITEM34
001550  056233                          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
001552  063406                          DH36            ;PC
                                                        ;PC OF JSR
                                                        ;TEST NUMBER
                                                        ;RHMR
                                                        ;POSITION REG.
```

```
                                                        ;PATTERN REGISTER

001554  064504                          DT36          ;$ERRPC,PCJSR,TSTNM,MR,EC1,EC2

001556  064755                          DF36          ;0,0,0,0,0,0


                                ;ITEM35
001560  056326                          EM35          ;POSITION REGISTER OR 11 BITS OF
                                                      ;PATTERN REGISTER INCORRECT
                                                      ;LOWER 11 BITS OF PATTERN REGISTER
                                                      ;SHOULD MATCH LOWER 11 BITS OF GOOD ECC1
                                                      ;DATA ENVELOPE AND N-CODE ZEROS ARE IN DECIMAL
001562  063545                          DH37          ;PC
                                                      ;TEST NUMBER
                                                      ;ECC POSITION
                                                      ;GOOD POSITION
                                                      ;GOOD ECC1
                                                      ;GOOD ECC2
                                                      ;ECC PATTERN
                                                      ;DATA ENVELOPE
                                                      ;N-CODE ZEROS

001564  064522                          DT37          ;$ERRPC,TSTNM,EC1,POSITI,GECC1,GECC2,EC2,DATENV,ZCODE

001566  064763                          DF37          ;0,0,0,0,0,0,0,0,0,0


                                ;ITEM36
001570  056625                          EM36          ;ON A READ COMMAND WITH NON CORRECTABLE
                                                      ;ERROR INSERTED DCK AND ECH SHOULD BE SET
001572  063203                          DH35          ;PC
                                                      ;TEST NUMBER
                                                      ;GOOD ECC1
                                                      ;GOOD ECC2
                                                      ;PATTERN REGISTER
                                                      ;POSITION REGISTER
                                                      ;RHER1

001574  064462                          DT35          ;$ERRPC,TSTNM,GECC1,GECC2,EC2,EC1,ER1

001576  064745                          DF35          ;0,0,0,0,0,0,0


                                ;ITEM37
001600  056736                          EM37          ;PGE ERROR
001602  057727                          DH11          ;PC
                                                      ;TEST NO
                                                      ;FAILING REG. ADDR
                                                      ;CONTENTS OF RHCS1
                                                      ;CONTENTS OF RHCS2
                                                      ;CONTENTS OF RHDS1
```

```
                                                        ;CONTENTS OF RHER1
001604  064102                          DT11            ;$ERRPC,$TSTNM,$BDADR,CS1,CS2,DS1,ER1
001606  064574                          DF11            ;0,0,0,0,0,0
                                ;ITEM40
001610  057073                          EM40            ;RHWC DID NOT = 0 AFTER A READ OR
                                                        ;WRITE HEADER AND DATA
001612  063763                          DH40            ;PC
                                                        ;TEST NO
                                                        ;CONTENTS OF RHWC
001614  064546                          DT40            ;$ERRPC,TSTNM,$BDDAT
001616  064774                          DF40            ;0,0,0
```

```
                        ;*****************************************************************
                        ;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)
```

```
776             002000                  PSEL=   2000                    ;PORT SELECT (BIT #10)
777             004000                  DVA=    4000                    ;DEVICE AVAILABLE (BIT #11)
778             020000                  MCPE=   20000                   ;MASSBUSS PARITY ERROR (BIT #13)
779             040000                  TRE=    40000                   ;TRANSFER ERROR (BIT #14)
780             100000                  SC=     100000                  ;SPECIAL CONDITION (BIT #15)
781
782                                     ;STATUS REGISTER (RHDS1) (#01)
783
784             000001                  DFS=    1                       ;DRIVE FORWARD 5"/SEC.  (BIT #0)
785             000002                  DFF20=  2                       ;DRIVE FORWARD 20"/SEC.  (BIT #1)
786             000004                  DIGB=   4                       ;DRIVE TO INNER GAVRD BAND (BIT #2)
787             000010                  GRV=    10                      ;GO REVERSE (BIT #3)
788             000020                  DL64=   20                      ;DIFFERENCE LESS THAN 64 (BIT #4)
789             000040                  DE1=    40                      ;DIFFERENCE EQUALS 1 (BIT #5)
790             000100                  VV=     100                     ;VOLUME VALID (BIT #6)
791             000200                  DRY=    200                     ;DRIVE READY (BIT #7)
792             000400                  DPR=    400                     ;DRIVE PRESENT (BIT #8)
793             001000                  PROG=   1000                    ;PROGRAMABLE (BIT #9)
794             002000                  LST=    2000                    ;LAST SECTOR TRANSFERRED (BIT #10)
795             004000                  WRL=    4000                    ;WRITE LOCK (BIT #11)
796             010000                  MOL=    10000                   ;MEDIUM ON-LINE (BIT #12)
797             020000                  PIP=    20000                   ;POSITIONING OPERATION IN PROGRESS (BIT #13)
798             040000                  ERR=    40000                   ;COMPOSIT ERROR. (BIT #14)
799             100000                  ATA=    100000                  ;ATTENTION ACTIVE (BIT #15)
800
801                                     ;ERROR REGISTER #01 (RHER1) (#02)
802             000001                  ILF=    1                       ;ILLEGAL FUNCTION (BIT #0)
803             000002                  ILR=    2                       ;ILLEGAL REGISTER (BIT #1)
804             000004                  RMR=    4                       ;REGISTER MODIFICATION REFUSED (BIT #2)
805             000010                  PAR=    10                      ;PARITY ERROR (BIT #3)
806             000020                  FER=    20                      ;FORMAT ERROR (BIT #4)
807             000040                  WCF=    40                      ;WRITE CLOCK FAIL (BIT #5)
808             000100                  ECH=    100                     ;ECC HARD ERROR (BIT #6)
809             000200                  HCE=    200                     ;HEADER COMPARE ERROR (BIT #7)
810             000400                  HCRC=   400                     ;HEADER CRC ERROR (BIT #8)
811             001000                  AOE=    1000                    ;ADDRESS OVERFLOW ERROR (BIT #9)
812             002000                  IAE=    2000                    ;INVALID ADDRESS ERROR (BIT #10)
813             004000                  WLE=    4000                    ;WRITE LOCK ERROR (BIT #11)
814             010000                  DTE=    10000                   ;DRIVE TIMING ERROR (BIT #12)
815             020000                  OPI=    20000                   ;OPERATION INCOMPLETE (BIT #13)
816             040000                  UNS=    40000                   ;DRIVE UNSAFE (BIT #14)
817             100000                  DCK=    100000                  ;DATA CHECK ERROR (BIT 15)
818
819                                     ;MAINTAINABILITY REGISTER (RHMR)(#03)
820
821             000001                  DMD=    1                       ;DIAGINOSTIC MODE (BIT #0)
822             000002                  MCLK=   2                       ;MAINTAINABILITY CLOCK (BIT #1)
823             000004                  MINX=   4                       ;MAINTAINABILITY INDEX (BIT #2)
824             000010                  MSTCK=  10                      ;MAINTAINABILITY SECTOR CLOCK (BIT #3)
825             000020                  MRD=    20                      ;MAINTAINABILITY READ (BIT #4)
826             000040                  MWR=    40                      ;MAINTAINABILITY WRITE (BIT #5)
827             000200                  DENVL=  200                     ;DATA ENVELOPE (BIT #7)
828             000400                  ZER=    400                     ;ZERO DETECT (BIT #8)
829             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)
000004                          AT2=    4               ;DEVICE 2 (BIT #2)
000010                          AT3=    10              ;DEVICE 3 (BIT #3)
000020                          AT4=    20              ;DEVICE 4 (BIT #4)
000040                          AT5=    40              ;DEVICE 5 (BIT #5)
000100                          AT6=    100             ;DEVICE 6 (BIT #6)
000200                          AT7=    200             ;DEVICE 7 (BIT #7)



                                ;DESIRED SECTOR/TRACK ADDRESS REGISTER (RHDST) (#1)
                                ;EACH BIT IS CALLED BY BIT NUMBER



                                ;DRIVE TYPE REGISTER (RHDT) (#06)
                                ;EACH BIT IS CALLED BY BIT NUMBER



                                ;LOOK-AHEAD REGISTER (RHLA) (#07)

000001                          EXT1=   1               ;EXTENSION 1 (BIT #0)
000002                          EXT2=   2               ;EXTENSION 2 (BIT #1)
000004                          EXT4=   4               ;EXTENSION 3 (BIT #2)
000010                          EXT10=  10              ;EXTENSION 4 (BIT #3)
000020                          EXT20=  20              ;EXTENSION 5 (BIT #4)
000040                          EXT40=  40              ;EXTENSION 6 (BIT #5)
000100                          SC1=    100             ;SECTOR COUNT FIELD 0 (BIT #6)
000200                          SC2=    200             ;SECTOR COUNT FIELD 1 (BIT #7)
000400                          SC4=    400             ;SECTOR COUNT FIELD 2 (BIT #8)
001000                          SC10=   1000            ;SECTOR COUNT FIELD 3 (BIT #9)
002000                          SC20=   2000            ;SECTOR COUNT FIELD 4 (BIT #10)
004000                          TRK1=   4000            ;TRACK FIELD 1 (BIT #11)
010000                          TRK2=   10000           ;TRACK FIELD 2 (BIT #12)
020000                          TRK4=   20000           ;TRACK FIELD 3 (BIT #13)
040000                          TRK10=  40000           ;TRACK FIELD 4 (BIT #14)
100000                          TRK20=  100000          ;TRACK FIELD 5 (BIT #15)

                                ;ERROR REGISTER #2 (RHER2) (#10)

000001                          WCU=    1               ;WRITE CURRENT UNSAFE (BIT #0)
000002                          CSF=    2               ;CURRENT SINK FAILURE (BIT #1)
000004                          WSU=    4               ;WRITE SELECT UNSAFE (BIT #2)
```

```
                000010          CSU=    10              ;CURRENT SWITCH UNSAFE (BIT #3)
                000020          MSE=    20              ;MOTOR SEQUENCE ERROR (BIT #4)
                000040          TDF=    40              ;TRANSITIONS DETECTOR FAILURE (BIT #5)
                000100          TUF=    100             ;TRANSITIONS UNSAFE (BIT #6)
                000200          FEN=    200             ;FAI_SAFE ENABLED (BIT #7)
                000400          WRU=    400             ;WRITE READY UNSAFE (BIT #8)
                001000          MHS=    1000            ;MULTIPLE HEAD SELECT (BIT #9)
                002000          NHS=    2000            ;NO HEAD SELECTION (BIT #10)
                004000          IXE=    4000            ;INDEX ERROR (BIT #11)
                010000          VU30=   10000           ;30VOLT UNSAFE (BIT #12)
                020000          PLU=    20000           ;PLO UNSAFE (BIT #13)
                100000          ACU=    100000          ;ACUNSAFE (BIT #15)

                                ;OFFSET REGISTER (RHOF) (#11)

                000001          OF25=   1               ;OFFSET 25 MICRO INCHES (BIT #0)
                000002          OF50=   2               ;OFFSET 50 MICRO INCHES (BIT #1)
                000004          OF100=  4               ;OFFSET 100 MICRO INCHES (BIT #2)
                000010          OF200=  10              ;OFFSET 200 MICRO INCHES (BIT #3)
                000020          OF400=  20              ;OFFSET 400 MICRO INCHES (BIT #4)
                000040          OF800=  40              ;OFFSET 800 MICRO INCHES (BIT #5)

                000200          OFREV=  200             ;OFFSET NEGATIVE (REVERSE) (BIT #7)
                002000          HCI=    2000            ;HEADER COMPARE INHIBIT (BIT #10)
                004000          ECI=    4000            ;ERROR CORRECTION CODE INHIBIT (BIT #11)
                010000          FMT22=  10000           ;FORMAT BIT (BIT #12


                                ;DESIRED CYLINDER ADDRESS (RHCA) (#12)
                                ;EACH BIT IS CALLED BY BIT NUMBER.



                                ;CURRENT CYLINDER ADDRESS (RHCC) (#13)
                                ;EACH BIT IS CALLED BY BIT NUMBER



                                ;SERIAL NUMBER REGISTER (RHSN) (#14)
                                ;EACH IS CALLED BY BIT NUMBER



                                ;ERROR REGISTER #03 (RHER3) (#15)
```

```
                                PSU=    1               ;PACK SPEED UNSAFE (BIT #0)
                                VUF=    2               ;VELOCITY UNSAFE (BIT #1)
                                UWR=    10              ;ANY UNSAFE EXCEPT READ/WRITE (BIT #3)
                                PRE=    20              ;DISK PACK ROTATION ERROR (BIT #4)
                                ACL=    40              ;AC LOW (BIT #5)
                                DCL=    100             ;DC LOW (BIT #6)
                                SKI=    40000           ;SEEK INCOMPLETE (BIT #14)
                                OCYL=   100000          ;OFF CYLINDER (BIT #15)
                                .



                                ;ECC POSITION REGISTER (RHEC1) (#16)
                                ;EACH BIT IS CALLED BY BIT NUMBER



                                ;ECC PATTERN REGISTER (RHEC2) (#17)
                                ;EACH BIT IS CALLED BY BIT NUMBER
```

```
                                .SBTTL  REGISTER ADDRESSES



                                ;RP04 VECTOR ADDRESS
001620  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"

001622  176722                  RHDB:   176722                  ;DATA BUFFER SEE NOTE ABOVE
001624  176702                  RHWC:   176702                  ;WORD COUNT SEE NOTE ABOVE
001626  176704                  RHBA:   176704                  ;BUS ADDRESS SEE NOTE ABOVE
001630  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"

001632  176700                  RHCS1:  176700                  ;CONTROL AND STATUS 1 SEE NOTE ABOVE
001634  176714                  RHER1:  176714                  ;ERROR #1 SEE NOTE ABOVE
001636  176706                  RHDST:  176706                  ;DESIRED SECTOR/TRACK ADDRESS SEE NOTE ABOVE
001640  176740                  RHER2:  176740                  ;ERROR #2 SEE NOTE ABOVE
001642  176732                  RHOF:   176732                  ;OFFSET SEE NOTE ABOVE
001644  176734                  RHCA:   176734                  ;DESIRED CYLINDER ADDRESS SEE NOTE ABOVE
001646  176742                  RHER3:  176742                  ;ERROR #3 SEE NOTE ABOVE
001650  176716                  RHAS:   176716                  ;ATTENTION SUMMARY SEE NOTE ABOVE
001652  176724                  RHMR:   176724                  ;MAINTAINABILITY SEE NOTE ABOVE
001654  176712                  RHDS1:  176712                  ;DRIVE STATUS SEE NOTE ABOVE
001656  176726                  RHDT:   176726                  ;DRIVE TYPE SEE NOTE ABOVE
001660  176730                  RHSN:   176730                  ;SERIAL NUMBER SEE NOTE ABOVE
001662  176744                  RHEC1:  176744                  ;ECC POSITION SEE NOTE ABOVE
001664  176746                  RHEC2:  176746                  ;ECC PATTERN SEE NOTE ABOVE
001666  176720                  RHLA:   176720                  ;LOOK-AHEAD SEE NOTE ABOVE
001670  176736                  RHCC:   176736                  ;CURRENT CYLINDER ADDRESS SEE NOTE ABOVE
```

```
1009
1010
1011                                    ;THE FOLLOWING LOCATIONS ARE RESERVED FOR REGISTER SAVES
1012                                    ;ANY TIME THERE IS AN ERROR ALL THESE WILL BE FILLED
1013                                    ;ONLY SOME MAY BE PRINTED BUT ALL WILL BE FILLED TRUE
1014                                    ;FOR THE TIME JUST AFTER THE "ERROR" ERROR COMMAND
1015
1016  001672  000000                    DB:     0                       ;DATA BUFFER
1017  001674  000000                    WC:     0                       ;WORD COUNT
1018  001676  000000                    BA:     0                       ;BUS ADDRESS
1019  001700  000000                    CS2:    0                       ;CONTROL AND STATUS 2
1020
1021
1022  001702  000000                    CS1:    0                       ;CONTROL AND STATUS 1
1023  001704  000000                    ER1:    0                       ;ERROR #1
1024  001706  000000                    DST:    0                       ;DESIRED SECTOR/TRACK ADDRESS
1025  001710  000000                    ER2:    0                       ;ERROR #2
1026  001712  000000                    OF:     0                       ;OFFSET
1027  001714  000000                    CA:     0                       ;DESIRED CYLINDER ADDRESS
1028  001716  000000                    ER3:    0                       ;ERROR #3
1029  001720  000000                    AS:     0                       ;ATTENTION SUMMARY
1030  001722  000000                    MR:     0                       ;MAINTAINABILITY
1031  001724  000000                    DS1:    0                       ;DRIVE STATUS
1032  001726  000000                    DT:     0                       ;DRIVE TYPE
1033  001730  000000                    SN:     0                       ;SERIAL NUMBER
1034  001732  000000                    EC1:    0                       ;ECC POSITION
1035  001734  000000                    EC2:    0                       ;ECC PATTERN
1036  001736  000000                    LA:     0                       ;LOOK-AHEAD
1037  001740  000000                    CC:     0                       ;CURRENT CYLINDER ADDRESS
```

```
1038
1039                                    ;FLAGS & INTERNAL PROGRAM CONTROL WORDS
1040
1041  001742  000010                    UNITS:  .BLKW   8.              ;TABLE OF DRIVES PRESENT TO TEST
1042  001762  000000                    UNIT:   .WORD   0               ;UNIT UNDER TEST
1043  001764  000000                    NOUNIT: .WORD   0               ;NUMBER OF UNITS PRESENT
1044                                                                    ;USED TO KEEP TRACK OF UNIT UNDER TEST
1045  001766  000000                    NUNIT:  .WORD   0               ;USED TO DETERMINE IF THERE IS MORE
1046                                                                    ;THAN ONE UNIT
1047  001770  000000                    SELECT: .WORD   0               ;ALL ONES INDICATE UNIT TO BE SELECTED
1048  001772  000000                    UNITSL: .WORD   0               ;UNIT NO. SELECTED
1049
1050  001774  000000                    ERFLGS: 0                       ;ERROR FLAG
1051
1052  001776  000000                    SAVDT:  0                       ;SAVE DRIVE TYPE REGISTER
1053                                                                    ;FOR COMPARISON IN DRIVE CLEAR TEST
1054                                                                    ;AND RH INIT TEST
1055  002000  000000                    SAVSN:  0                       ;SAVE SERIAL NUMBER REGISTER
1056                                                                    ;FOR COMPARISON IN DRIVE CLEAR TEST
1057                                                                    ;AND RH INIT TEST
1058
1059  002002  000000                    PCJSR:  0                       ;SAVE PC OF JSR WHICH GAVE THE ERROR
1060
1061  002004  000000                    ATTENT: 0                       ;ATTENTION BIT FOR PRESENT UNIT
1062  002006  000000                    TOTALAT: 0                      ;TATAL ATTENTION BITS
1063
1064  002010  000000                    TMPILL: 0                       ;TEMPORARY ILLEGAL FUNCTION
1065
1066  002012  000000                    TSECC:  0                       ;FLAG TO SAY IF ECC TEST OR NOT
1067                                                                    ;WHEN =177777 IT IS AN ECC TEST
1068                                                                    ;WHEN =0IT IS NOT AN ECC TEST
1069
1070  002014  000000                    TESDTE: 0                       ;FLAG TO SAY IF DRIVE TIMING ERROR OR NOT
1071                                                                    ;WHEN = 177777 IT IS A DTE TEST
1072                                                                    ;WHEN = 0      IT IS NOT A DTE TEST
1073
1074  002016  000000                    TAGDTE: 0                       ;TEMPORARY TAG USED IN DRIVE TIMING
1075                                                                    ;ERROR TEST
1076
1077  002020  000000                    TSTNM:  0                       ;TEST NUMBER
1078
1079  002022  000000                    FIRST:  0                       ;IF ZERO WILL TYPE HEADER
```

```
1080
1081
1082                                    ;FUNCTION EQUATES
1083
1084                                    ;TABLE OF FUNCTIONS FOR RHCS1, THEN "GO" BIT HAS TO BE SET
1085
1086  002024                    FUTABL:
1087  002024  000000           NOPERA: 0                       ;NO OPERATION
1088  002026  000002           UNLOAD: 2                       ;UNLOAD (STAND BY)
1089  002030  000006           RECALI: 6                       ;RECALIBRATE
1090  002032  000010           DCLEAR: 10                      ;DRIVE CLEAR
1091  002034  000012           RELEAS: 12                      ;RELEASE (DUAL-PORT OPERATION)
1092  002036  000030           SERCH:  30                      ;SEARCH COMMAND
1093  002040  000050           WRCHEK: 50                      ;WRITE CHECK DATA
1094  002042  000052           WRCHDT: 52                      ;WRITE CHECK HEADER AND DATA
1095  002044  000060           WRIDAT: 60                      ;WRITE DATA
1096  002046  000062           WRIFOR: 62                      ;WRITE HEADER AND DATA (FORMAT)
1097  002050  000070           READAT: 70                      ;READ DATA
1098  002052  000072           REFOR:  72                      ;READ HEADER AND DATA
1099  002054  000004           SEECOM: 4                       ;SEEK COMMAND
1100  002056  000014           OFSETC: 14                      ;OFFSET COMMAND
1101  002060  000016           RETCL:  16                      ;RETURNTO CENTERLINE
1102  002062  000022           PKACK:  22                      ;PACK ACKNOWLEDGE
1103  002064  000020           READIN: 20                      ;READ IN
1104  002066  000000           ILLEGL: .WORD                   ;COMPUTED ILLEGAL FUNCTION
1105
1106
1107                                    ;DATA BUFFER FOR READ WRITE
1108
1109  002070  000422                    WRFROM: .BLKW   274.            ;WRITE FROM THIS BUFFER
1110  003134  000422                    REINTO: .BLKW   274.            ;READ INTO THIS BUFFER
1111
1112
1113
1114                                    ;TABLE FOR ATTENTION BITS
1115                                    ;ATTENTION TABLE
1116
1117  004200    001     002     004   ATABLE: .BYTE   1,2,4,10,20,40,100,200
1118  004203    010     020     040
1119  004206    100     200
1120
```

```
1121
1122                                            .SBTTL
1123                                            .SBTTL  ***DIAGNOSTIC CODE***
1124                                            .SBTTL
1125
1126                                            .SBTTL  SETUP TESTS
1127
1128  004210  012737  177777  001770  BEGIN2: MOV     #-1,@#SELECT    ;SELECT UNIT
1129  004216  000402                  BR      START
1130  004220  005037  001770          BEGIN:  CLR     @#SELECT        ;DO NOT SELECT UNIT
1131                                                                  ;NORMAL RUN
1132
1133  004224                          START:
1134  004224  012737  000340  177776          MOV     #340,@#PS       ;;LOCK OUT ALL INTERRUPTS
1135  004232  012706  001100                  MOV     #$CMTAG,R6      ;;FIRST LOCATION TO BE CLEARED
1136  004236  005026                          CLR     (R6)+           ;;CLEAR MEMORY LOCATION
1137  004240  022706  001136                  CMP     #STKS,R6        ;;DONE?
1138  004244  001374                          BNE     .-6             ;;LOOP BACK IF NO
1139  004246  012706  001000                  MOV     #STACK,SP       ;;SETUP THE STACK POINTER
1140  004252  012737  051652  000020          MOV     #$SCOPE,@#IOTVEC;;IOT VECTOR FOR SCOPE ROUTINE
1141  004260  012737  000340  000022          MOV     #340,@#IOTVEC+2 ;;LEVEL 7
1142  004266  012737  053240  000030          MOV     #$ERROR,@#EMTVEC;;EMT VECTOR FOR ERROR ROUTINE
1143  004274  012737  000340  000032          MOV     #340,@#EMTVEC+2 ;;LEVEL 7
1144  004302  012737  054004  000034          MOV     #$TRAP,@#TRAPVEC;;TRAP VECTOR FOR TRAP CALLS
1145  004310  012737  000340  000036          MOV     #340,@#TRAPVEC+2;LEVEL 7
1146  004316  012737  054052  000024          MOV     #$PWRDN,@#PWRVEC ;;POWER FAILURE VECTOR
1147  004324  012737  000340  000026          MOV     #340,@#PWRVEC+2 ;;LEVEL 7
1148  004332  005067  174646                  CLR     $TIMES          ;;INITIALIZE NUMBER OF ITERATIONS
1149  004336  005067  174644                  CLR     $ESCAPE         ;;CLEAR THE ESCAPE ON ERROR ADDRESS
1150  004342  112767  000001  174545          MOVB    #1,$ERMAX       ;;ALLOW ONE ERROR PER TEST
1151  004350  012767  004350  174530          MOV     #.,$LPADR       ;;INITIALIZE THE LOOP ADDRESS FOR SCOPE
1152  004356  012767  004356  174524          MOV     #.,$LPERR       ;;SETUP THE ERROR LOOP ADDRESS
1153
1154
1155
1156  004364  012767  000000  173404          MOV     #0,PS           ;SET PROCESSOR STATUS TO 0
1157  004372  012777  051570  175220          MOV     #RPVECT,@RPVEC  ;THIS IS FOR UNTIMELY RP04 INTERRUPTS
1158  004400  004737  052512                  JSR     PC,@#STKINT     ;INITILIZE THE TTY KEYBOARD
1159  004404  005737  002022                  TST     @#FIRST         ;IS THIS FIRST TIME ROUND
1160  004410  001001                          BNE     1$              ;SKIP HEADER IF NOT
1161  004412  000402                          BR      2$              ;DO HEADER IF SO
1162  004414  000137  005240          1$:     JMP     @#SND1          ;SKIP OVERALL PROGRAM HEADER
1163
1164  004420                          2$:
1165  004420  104400  004426                  TYPE    ,.+4            ;;TYPE ASCIZ STRING
1166  004424  000435                          BR      64$             ;;GET OVER THE ASCIZ
1167                                          ;;.ASCIZ        <15><12>/RJP04 DISKLESS RH11 TEST - PART I (STATIC 1A)- DZRPS-D/
1168  004520                          64$:
1169  004520  104400  004526                  TYPE    ,.+4            ;;TYPE ASCIZ STRING
1170  004524  000416                          BR      65$             ;;GET OVER THE ASCIZ
1171                                          ;;.ASCIZ        <15><12>/REVISION DATE: 21-JULY-75/
1172  004562                          65$:
1173  004562  104400  004570                  TYPE    ,.+4            ;;TYPE ASCIZ STRING
1174  004566  000402                          BR      66$             ;;GET OVER THE ASCIZ
```

```
1175                                                    ;;.ASCIZ    <15><12>/ /
1176  004574                                    66$:
1177  004574  104400  004602                           TYPE    ,.+4            ;;TYPE ASCIZ STRING
1178  004600  000432                                   BR      67$             ;;GET OVER THE ASCIZ
1179                                                    ;;.ASCIZ    <15><12>/ALL DCL UNDER TEST MUST BE LOCKED ON CORRECT PORT/
1180  004666                                    67$:
1181  004666  104400  004674                           TYPE    ,.+4            ;;TYPE ASCIZ STRING
1182  004672  000427                                   BR      68$             ;;GET OVER THE ASCIZ
1183                                                    ;;.ASCIZ    <15><12>/IF CHANGES ARE REQUIRED ON PORT SWITCH THEN/
1184  004752                                    68$:
1185  004752  104400  004760                           TYPE    ,.+4            ;;TYPE ASCIZ STRING
1186  004756  000430                                   BR      69$             ;;GET OVER THE ASCIZ
1187                                                    ;;.ASCIZ    <15><12>/A CYCLE UP SEQUENCE IS REQUIRED FOR STROBING/
1188  005040                                    69$:
1189  005040  104400  005046                           TYPE    ,.+4            ;;TYPE ASCIZ STRING
1190  005044  000414                                   BR      70$             ;;GET OVER THE ASCIZ
1191                                                    ;;.ASCIZ    <15><12>/THE PORT SELECT FLOP/
1192  005076                                    70$:
1193  005076  104400  005104                           TYPE    ,.+4            ;;TYPE ASCIZ STRING
1194  005102  000402                                   BR      71$             ;;GET OVER THE ASCIZ
1195                                                    ;;.ASCIZ    <15><12>/ /
1196  005110                                    71$:
1197  005110  104400  005116                           TYPE    ,.+4            ;;TYPE ASCIZ STRING
1198  005114  000427                                   BR      72$             ;;GET OVER THE ASCIZ
1199                                                    ;;.ASCIZ    <15><12>/ALL DCL NOT UNDER TEST MUST BE SWITCHED OFF/
1200  005174                                    72$:
1201  005174  104400  005202                           TYPE    ,.+4            ;;TYPE ASCIZ STRING
1202  005200  000417                                   BR      73$             ;;GET OVER THE ASCIZ
1203                                                    ;;.ASCIZ    <15><12>/OR LOCKED ON THE OTHER PORT/
1204  005240                                    73$:
1205
1206  005240  012737  177777  002022  SND1:    MOV     #-1,@#FIRST     ;NEXT TIME DO NOT GIVE HEADER
1207  005246  005737  001770                   TST     @#SELECT        ;WAS IT A 200 START
1208  005252  001435                           BEQ     TST1            ;BRANCH IF STARTING FROM 200
1209
1210  005254  104400  005262                   TYPE    ,.+4            ;;TYPE ASCIZ STRING
1211  005260  000423                           BR      64$             ;;GET OVER THE ASCIZ
1212                                                    ;;.ASCIZ    <15><12>/SELECT UNIT NUMBER TO BE TESTED ?  /
1213  005330                                    64$:
1214  005330  104416                           RDOCT
1215  005332  042716  177770                   BIC     #177770,(SP)    ;ONLY KEEP LAST 3 BITS
1216  005336  011637  001762                   MOV     (SP),@#UNIT     ;SAVE UNIT TO BE TESTED
1217  005342  012637  001772                   MOV     (SP)+,@#UNITSL  ;SAVE UNIT TO BE TESTED
1218
1219
```

```
1220
1221                                    ;*****************************************************************
1222                                    ;*TEST 1         REFERENCE EACH REGISTER
1223                                    ;*              REFERENCE EACH REGISTER BY A MOVE INSTRUCTION
1224                                    ;*****************************************************************
1225  005346  000004                    TST1:   SCOPE
1226  005350  012767  000001  173626            MOV     #1,$TIMES       ;;DO 1 ITERATION
1227  005356  012706  001000                    MOV     #STACK, SP      ;SET UP STACK POINTER
1228  005362  012737  000001  002020            MOV     #TTNO,@#TSTNM   ;THIS SAVES TEST NUMBER
1229  005370  012737  053246  000030            MOV     #REGSAI,@#EMTVEC;ERROR VECTOR SO THAT
1230                                                                    ;NO REGISTERS ARE SAVED
1231  005376  012737  005424  000004            MOV     #2$,    @#ERRVEC ;SET UP FOR BUS TIMEOUT
1232  005404  012700  000024                    MOV     #24,    R0      ;THERE ARE 24 REG TO TEST
1233  005410  012701  001622                    MOV     #RH0B,  R1      ;R1 NOW HAS ADDR OF ADDR OF FIRST REG.
1234  005414  013102                    1$:     MOV     @(R1)+, R2      ;READ HARDWARE REG.
1235  005416  005300                            DEC     R0              ;COUNT DOWN
1236  005420  001375                            BNE     1$              ;BRANCH IF 24 NOT DONE
1237  005422  000471                            BR      3$              ;BRANCH IF 24 DONE
1238  005424  012737  000006  000004    2$:     MOV     #ERRVEC+2,@#ERRVEC ;RESTORE TRAP CATCHER
1239  005432  022626                            CMP     (SP)+,  (SP)+   ;CLEAN STACK
1240  005434  016167  177776  173530            MOV     -2(R1), $TMP1   ;STORE FAILING REG ADDR
1241  005442  104015                            ERROR   15              ;REGISTER NON EXISTANT
1242  005444  032737  020000  177570            BIT     #SW13,@#SWR     ;INHIBIT ERROR PRINTOUT ?
1243  005452  001053                            BNE     4$              ;BRANCH IF YES
1244
1245  005454  104400  005462                    TYPE    ,.+4            ;;TYPE ASCIZ STRING
1246  005460  000431                            BR      64$             ;;GET OVER THE ASCIZ
1247                                            ;;.ASCIZ        <15><12>/IF BASE ADDRESS IS TO BE CHANGED HALT PROGRAM /
1248  005544                            64$:
1249  005544  104400  005552                    TYPE    ,.+4            ;;TYPE ASCIZ STRING
1250  005550  000411                            BR      65$             ;;GET OVER THE ASCIZ
1251                                            ;;.ASCIZ        <15><12>/AND RESTART AT /
1252  005574                            65$:
1253
1254  005574  012746  044146                    MOV     #BASECH,-(SP)   ;GET READY TO TYPE STARTING ADDRESS
1255                                                                    ;OF "CHANGE OF BASE ADDRESS" ROUTINE
1256  005600  104402                            TYPOC
1257  005602  000137  037204            4$:     JMP     @#$EOP          ;GO TO END OF PROGRAM
1258  005606  012737  053240  000030  3$:      MOV     #$ERROR,@#EMTVEC;RESTORE ERROR VECTOR
1259                                                                    ;SO THAT REGISTERS ARE SAVED
1260  005614  012737  000006  000004            MOV     #ERRVEC+2,@#ERRVEC ;RESTORE TRAP CATCHER
1261
```

```
1262
1263
1264
1265                                    ;*****************************************************************
1266                                    ;*TEST 2        RHCS2-CONTROL AND STATUS 2
1267
1268                                    ;*      THIS PARTIALLY TESTS RHCS2 TO ENABLE DETERMINATION
1269                                    ;*      OF THE NUMBER OF DRIVES PRESENT
1270
1271                                    ;*****************************************************************
1272  005622  000004                    TST2:   SCOPE
1273  005624  012767  000001  173352            MOV     #1,$TIMES       ;;DO 1 ITERATION
1274  005632  012706  001000                    MOV     #STACK,SP       ;RESET STACK
1275  005636  012737  000002  002020            MOV     #TTNO,@#TSTNM   ;THIS SAVES TEST NUMBER
1276
1277  005644  013737  010126  005664            MOV     @#PRCS2+12,@#UN
1278  005652  013737  001630  005666            MOV     @#RHCS2,@#UN+2
1279  005660  004537  040174                    JSR     R5,@#BITST      ;TEST BITS IN REGISTER
1280  005664  000000                    UN:     .WORD   0               ;ONLY THESE BITS ARE TEST READ/WRITE
1281  005666  000000                            .WORD   0               ;ADDRESS OF REG. BEING TESTED
1282  005670  104001                            ERROR   1               ;IN CORRECT DATA RECEIVED
1283  005672  000207                            RTS     PC              ;RETURN TO BLT3 ROUTINE
1284
1285
1286
1287
1288
1289
1290                                    ;*****************************************************************
1291                                    ;*TEST 3        PARTIAL TEST FOR RHAS FOR UNIT NUMBERS PRESENT
1292                                    ;*****************************************************************
1293  005674  000004                    TST3:   SCOPE
1294  005676  012767  000001  173300            MOV     #1,$TIMES       ;;DO 1 ITERATION
1295
1296  005704  012737  000003  002020            MOV     #TTNO,@#TSTNM   ;THIS SAVES TEST NUMBER
1297  005712  013701  001650                    MOV     @#RHAS,R1       ;R1 HAS ADDRESS OF RHAS
1298  005716  012711  177777                    MOV     #-1,@R1         ;THIS CLEARS RHAS (SURPRISED!)
1299  005722  011137  001126                    MOV     @R1,@#$BDDAT    ;TEST DATA
1300  005726  105737  001126                    TSTB    @#$BDDAT
1301  005732  001405                            BEQ     TST4            ;BRANCH IF GOOD
1302  005734  005037  001124                    CLR     @#$GDDAT        ;GOOD DATA
1303  005740  010137  040172                    MOV     R1,@#REGADR     ;FAILING REG. RHAS
1304  005744  104001                            ERROR   1               ;RHAS DOES NOT CLEAR
1305                                                                    ;WITH ONES
```

```
1306
1307
1308                                    ;*****************************************************************
1309                                    ;*TEST 4        TEST FOR DRIVES PRESENT USING RHAS AND RHCS2
1310                                    ;*****************************************************************
1311  005746  000004                    TST4:   SCOPE
1312  005750  012767  000001  173226            MOV     #1,STIMES       ;;DO 1 ITERATION
1313
1314  005756  000005                            RESET                   ;START WITH AN INIT
1315  005760  004737  052512                    JSR     PC,@#STKINT     ;INITILIZE TTY KEYBOARD
1316  005764  032737  020000  177570            BIT     #SW13,@#SWR     ;INHIBIT ERROR TYPEOUT?
1317  005772  001155                            BNE     4$              ;BRANCH IF YES
1318
1319  005774  104400  006002                    TYPE    ,.+4            ;;TYPE ASCIZ STRING
1320  006000  000430                            BR      64$             ;;GET OVER THE ASCIZ
1321                                            ;;.ASCIZ    <15><12>/LOOKING AT RHAS - RP04 DRIVES ASSUMED PRESENT/
1322  006062                            64$:
1323  006062  104400  006070                    TYPE    ,.+4            ;;TYPE ASCIZ STRING
1324  006066  000402                            BR      65$             ;;GET OVER THE ASCIZ
1325                                            ;;.ASCIZ    <15><12>/ /
1326  006074                            65$:
1327  006074  104400  006102                    TYPE    ,.+4            ;;TYPE ASCIZ STRING
1328  006100  000430                            BR      66$             ;;GET OVER THE ASCIZ
1329                                            ;;.ASCIZ    <15><12>/*********************************************/
1330  006162                            66$:
1331  006162  104400  006170                    TYPE    ,.+4            ;;TYPE ASCIZ STRING
1332  006166  000424                            BR      67$             ;;GET OVER THE ASCIZ
1333                                            ;;.ASCIZ    <15><12>/THIS MUST BE VERIFIED BY OPERATOR !!/
1334  006240                            67$:
1335  006240  104400  006246                    TYPE    ,.+4            ;;TYPE ASCIZ STRING
1336  006244  000430                            BR      68$             ;;GET OVER THE ASCIZ
1337                                            ;;.ASCIZ    <15><12>/*********************************************/
1338  006326                            68$:
1339
1340  006326  013701  001650            4$:     MOV     @#RHAS,R1       ;LOAD R1 WITH ADDR. OF RHAS
1341  006332  013702  001630                    MOV     @#RHCS2,R2      ;LOAD R2 WITH ADDR. OF RHCS2
1342  006336  005012                            CLR     @R2             ;CLEAR RHCS2 (ADDRESS UNIT #0)
1343  006340  012700  000010                    MOV     #8.,R0          ;INITIALIZE DRIVE COUNTER
1344  006344  013704  001634                    MOV     @#RHER1,R4      ;LOAD R4 WITH ADDR. OF RHER1
1345  006350  012714  177777            1$:     MOV     #-1,@R4         ;MOVE ERRORS INTO RHER1 OF UNIT ADDRESSED
1346  006354  005212                            INC     @R2             ;INCREMENT UNIT NO. (RHCS2)
1347  006356  005300                            DEC     R0              ;COUNT DOWN DRIVE COUNTER
1348  006360  001373                            BNE     1$              ;TEST AND DO NEXT UNIT IF 8 NOT DONE
1349
1350  006362  111137  002006                    MOVB    @R1,@#TOTALAT   ;SAVE ALL RESULTING ATTENTION BITS
1351                                            ;(USED IN DRIVE CLEAR TEST)
1352  006366  105037  002007                    CLRB    @#TOTALAT+1     ;CLEAR UPPER BYTE
1353  006372  105711                            TSTB    @R1             ;TEST RHAS FOR ANY DRIVES PRESENT
1354  006374  001402                            BEQ     2$              ;NONE RESPONDING - TYPE THE MESSAGE
1355  006376  000167  000400                    JMP     XE2             ;SOME THERE - GO FILL "UNITS" TABLE
1356
1357  006402  032737  020000  177570    2$:     BIT     #SW13,@#SWR     ;INHIBIT ERROR TYPE OUT?
1358  006410  001402                            BEQ     3$              ;TYPE "NO DRIVES" MESSAGE IF NO
1359  006412  000167  000640                    JMP     SELTST          ;CHECK FOR SELECTED UNIT START AND LOAD
```

```
1360                                                    ;"UNITS" TABLE WITH DESIRED DRIVE IF SO
1361
1362  006416                                    3$:
1363  006416  104400  006424                            TYPE    .+4             ;;TYPE ASCIZ STRING
1364  006422  000412                            BR      69$             ;;GET OVER THE ASCIZ
1365                                            ;;.ASCIZ    <15><12>/NO DRIVES-RHAS=0/
1366  006450                                    69$:
1367  006450  104400  006456                            TYPE    .+4             ;;TYPE ASCIZ STRING
1368  006454  000402                            BR      70$             ;;GET OVER THE ASCIZ
1369                                            ;;.ASCIZ    <15><12>/ /
1370  006462                                    70$:
1371  006462  104400  006470                            TYPE    .+4             ;;TYPE ASCIZ STRING
1372  006466  000436                            BR      71$             ;;GET OVER THE ASCIZ
1373                                            ;;.ASCIZ    <15><12>/WRITING ONES INTO ERROR REGISTER #1 FOR ALL UNIT NUMBERS/
1374  006564                                    71$:
1375  006564  104400  006572                            TYPE    .+4             ;;TYPE ASCIZ STRING
1376  006570  000441                            BR      72$             ;;GET OVER THE ASCIZ
1377                                            ;;.ASCIZ    <15><12>/DOES NOT SET ANY BIT IN THE ATTENTION REGISTER SO ABORT P
1378  006674                                    72$:
1379  006674  104400  006702                            TYPE    .+4             ;;TYPE ASCIZ STRING
1380  006700  000436                            BR      73$             ;;GET OVER THE ASCIZ
1381                                            ;;.ASCIZ    <15><12>/TO LOOP ON THIS TEST WO PRINTOUT, SET SWITCHES 13, 8 & 2/
1382  006776                                    73$:
1383
1384  006776  000137  037204                    JMP     @#$EOP          ;GO OUT---------------------------------->
1385
1386  007002                                    XE2:
1387  007002  012700  000010                    2$:     MOV     #8.,R0          ;LOAD "UNITS" TABLE COUNTER
1388  007006  012703  001742                            MOV     #UNITS,R3       ;LOAD "UNITS" TABLE POINTER
1389  007012  012723  177777                    3$:     MOV     #-1,(R3)+       ;PRESET 1ST TABLE BLOCK TO ALL ONES
1390  007016  005300                            DEC     R0              ;COUNT DOWN
1391  007020  001374                            BNE     3$              ;PRESET NEXT BLOCK IF 8 NOT DONE
1392
1393  007022  012703  001742                    10$:    MOV     #UNITS,R3       ;RELOAD THE TABLE POINTER
1394  007026  005005                            CLR     R5              ;INITIALIZE UNIT NO. TO 0
1395  007030  005037  001764                            CLR     @#NOUNIT        ;NO. OF UNITS PRESENT
1396  007034  012700  000010                            MOV     #8.,R0          ;RELOAD THE TABLE COUNTER
1397  007040  011137  001170                            MOV     @R1,@#$TMP0     ;ADDR OF RHAS INTO TEMPORARY STORAGE
1398  007044  006037  001170                    4$:     ROR     @#$TMP0         ;SET CARRY IF 0 BIT = 1 (UNIT ATTEN.)
1399  007050  103067                            BCC     5$              ;CHECK NEXT UNIT IF ONE NOT IN BIT 0
1400
1401  007052  010577  172552                    11$:    MOV     R5,@RHCS2       ;INSERT UNIT NO. INTO RHCS2 UNIT ADDR.
1402  007056  022777  024020  172572                    CMP     #24020,@RHDT    ;READ RHDT - IS IT A DUAL PORT RP04 ?
1403  007064  001452                            BEQ     6$              ;YES...TYPE THE UNIT NO.
1404  007066  022777  020020  172562                    CMP     #20020,@RHDT    ;READ RHDT - IS IT A SINGLE PORT RP04 ?
1405  007074  001446                            BEQ     6$              ;YES...TYPE THE UNIT NO.
1406
1407                                            ;NO...IT'S NOT AN RP04 DEVICE SO TYPE
1408                                            ;OUT THE DEVICE TYPE
1409
1410  007076  104400  007104                            TYPE    .+4             ;;TYPE ASCIZ STRING
1411  007102  000410                            BR      64$             ;;GET OVER THE ASCIZ
1412                                            ;;.ASCIZ    <15><12>/UNIT NUMBER /
1413  007124                                    64$:
```

```
1414
1415  007124  010546                          MOV     R5,-(SP)        ;PUT THE UNIT NUMBER ON STACK
1416  007126  104410                          TYPDS                   ;TYPE IT
1417  007130  104400  007136                  TYPE    ,.+4            ;;TYPE ASCIZ STRING
1418  007134  000406                          BR      65$             ;;GET OVER THE ASCIZ
1419                                          ;;.ASCIZ      /,  RHDT = /
1420  007152                          65$:
1421  007152  017746  172500                  MOV     @RHDT,-(SP)     ;PUT RHDT ON THE STACK
1422  007156  104402                          TYPOC                   ;TYPE IT
1423  007160  104400  007166                  TYPE    ,.+4            ;;TYPE ASCIZ STRING
1424  007164  000411                          BR      66$             ;;GET OVER THE ASCIZ
1425                                          ;;.ASCIZ      / --- NOT AN RP04/
1426  007210                          66$:
1427  007210  000407                          BR      5$              ;UNIT NOT AN RP04 SO TEST NEXT ONE
1428  007212  010523                  6$:     MOV     R5,(R3)+        ;LOAD TABLE POSITION AND INCR IT
1429  007214  104400  001215                  TYPE    ,$CRLF          ;CRLF
1430  007220  010546                          MOV     R5,-(SP)        ;PUT RP04 UNIT NO. ON THE STACK
1431  007222  104410                          TYPDS                   ;TYPE THE UNIT NO.
1432  007224  005237  001764                  INC     @#NOUNIT        ;INCR THE TOTAL NO. OF UNITS
1433
1434  007230  005205                  5$:     INC     R5              ;'RHCS2' UNIT ADDRESS
1435  007232  005300                          DEC     R0              ;DRIVE COUNTER DOWN ONE
1436  007234  001303                          BNE     4$              ;TEST AND DO NEXT UNIT IF 8 NOT DONE
1437
1438  007236  013737  001742  001762  12$:     MOV     @#UNITS,@#UNIT  ;SET UNIT NO. TO FIRST ONE FOUND/OR 0
1439  007244  013737  001764  001766          MOV     @#NOUNIT,@#NUNIT ;SAVE NO. OF UNITS
1440  007252  005337  001766                  DEC     @#NUNIT         ;IF NUNIT = 0 THEN ONLY ONE UNIT
1441                                                                  ;IF NUNIT > 0 THEN MORE THAN ONE UNIT
1442  007256  005737  001770          SELTST: TST     @#SELECT        ;STARTING ADDRESS 200 ?
1443  007262  001403                          BEQ     TST5            ;BRANCH IF STARTING FROM 200
1444  007264  013737  001772  001762          MOV     @#UNITSL,@#UNIT ;CHANGE UNIT NUMBER TO SELECTED ONE
1445
```

```
1446
1447
1448                                    ;*****************************************************************
1449                                    ;*TEST 5        TEST SERIAL NUMBER AND DRIVE TYPEI
1450                                    ;*             READ SERIAL NUMBER REGISTER AND DRIVE TYPE REGISTER
1451                                    ;*             TYPE IT OUT AND PROCEED
1452                                    ;*             TO LOOP HERE SET SWITCH 8 AND THIS TEST NO AND RESTART
1453
1454                                    ;*****************************************************************
1455  007272  000004                    TST5:   SCOPE
1456  007274  012767  000001  171702            MOV     #1,$TIMES       ;;DO 1 ITERATION
1457  007302  012767  007504  171576            MOV     #1$,$LPADR      ;;SET SCOPE LOOP ADDRESS
1458
1459  007310  004737  040522                    JSR     PC,@#CLDISK     ;FILL UNIT NO.
1460  007314  005037  002004                    CLR     @#ATTENT        ;CLEAR
1461  007320  013700  001762                    MOV     @#UNIT,R0       ;R0 CONTAINS UNIT NO
1462  007324  116037  004200  002004          MOVB    ATABLE(R0),@#ATTENT ;SET APPROPIATE ATTENTION BIT
1463
1464  007332  104400  007340                    TYPE    ,.+4            ;;TYPE ASCIZ STRING
1465  007336  000415                            BR      64$             ;;GET OVER THE ASCIZ
1466                                            ;;.ASCIZ      <15><12>/TESTING DRIVE NUMBER  /
1467  007372                            64$:
1468
1469  007372  013746  001762                    MOV     @#UNIT,-(SP)    ;UNIT NO. TO STACK
1470  007376  104402                            TYPOC                   ;TYPE  DRIVE NO.
1471  007400  104400  001215                    TYPE    ,$CRLF
1472  007404  104400  007412                    TYPE    ,.+4            ;;TYPE ASCIZ STRING
1473  007410  000410                            BR      65$             ;;GET OVER THE ASCIZ
1474                                            ;;.ASCIZ      <15><12>/SERIAL NO. = /
1475  007432                            65$:
1476  007432  017746  172222                    MOV     @RHSN,-(SP)     ;;SAVE @RHSN FOR TYPEOUT
1477  007436  104402                            TYPOC                   ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
1478  007440  104400  001215                    TYPE    ,$CRLF
1479  007444  104400  007452                    TYPE    ,.+4            ;;TYPE ASCIZ STRING
1480  007450  000410                            BR      66$             ;;GET OVER THE ASCIZ
1481                                            ;;.ASCIZ      <15><12>/DRIVE TYPE = /
1482  007472                            66$:
1483  007472  017746  172160                    MOV     @RHDT,-(SP)     ;;SAVE @RHDT FOR TYPEOUT
1484  007476  104402                            TYPOC                   ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
1485  007500  104400  001215                    TYPE    ,$CRLF
1486  007504  005777  172150            1$:     TST     @RHSN           ;READ SERIAL NO. AND DRIVE TYPE
1487  007510  005777  172142                    TST     @RHDT           ;THESE TWO ARE TO HELP SCOPE LOOPS
1488  007514  017737  172140  002000            MOV     @RHSN,@#SAVSN   ;SAVE TO CHECK IF CLR RHCS2 BIT 5 CLEARS ANY BITS
1489  007522  017737  172130  001776            MOV     @RHDT,@#SAVDT   ;SAVE TO CHECK IF CLR RHCS2 BIT 5 CLEARS ANY BITS
1490
1491
```

```
1492
1493                                    ;*****************************************************************
1494                                    ;*TEST 6        CHECK MOL TO BE LOW
1495
1496                                    ;*      MAKE SURE THAT DRIVE IS OFF LINE BEFORE STARTING PROGRAM
1497                                    ;*      IF DRIVE IS ON LINE THEN AFTER TYPE OUT THE PROGRAM WILL
1498                                    ;*      HANG FOR EVER WAITING FOR DRIVE TO GO OFF LINE
1499
1500                                    ;*****************************************************************
1501  007530  000004                    TST6:   SCOPE
1502  007532  012737  000006  002020            MOV     #TTNO,@#TSTNM   ;THIS SAVES TEST NUMBER
1503  007540  004737  040522                    JSR     PC,@#CLDISK     ;GIVE INITILIZE
1504  007544  032713  010000                    BIT     #MOL,@R3        ;CHECK MOL IN RHDS1
1505  007550  001550                            BEQ     TST7            ;BRANCH IF MOL LOW
1506  007552  104400  007560                    TYPE    ,.+4            ;;TYPE ASCIZ STRING
1507  007556  000421                            BR      64$             ;;GET OVER THE ASCIZ
1508                                            ;;.ASCIZ        <15><12>/DRIVE IS ON LINE - MOL IS HIGH/
1509  007622                            64$:
1510  007622  104400  007630                    TYPE    ,.+4            ;;TYPE ASCIZ STRING
1511  007626  000424                            BR      65$             ;;GET OVER THE ASCIZ
1512                                            ;;.ASCIZ        <15><12>/HIT STOP ON DRIVE TO GET IT OFF LINE/
1513  007700                            65$:
1514  007700  104400  007706                    TYPE    ,.+4            ;;TYPE ASCIZ STRING
1515  007704  000430                            BR      66$             ;;GET OVER THE ASCIZ
1516                                            ;;.ASCIZ        <15><12>/PROGRAM WILL HANG TESTING MOL TILL MOL IS LOW/
1517  007766                            66$:
1518  007766  032713  010000            1$:     BIT     #MOL,@R3        ;CHECK MOL IN RHDS1
1519  007772  001375                            BNE     1$              ;BRANCH IF MOL IS HIGH
1520  007774  104400  010002                    TYPE    ,.+4            ;;TYPE ASCIZ STRING
1521  010000  000434                            BR      67$             ;;GET OVER THE ASCIZ
1522                                            ;;.ASCIZ        <15><12>/GOOD - MOL IS NOW LOW . PROGRAM WILL NOW BE EXECUTED/
1523  010072                            67$:
```

```
1524
1525                                            .SBTTL  REGISTER TESTS
1526
1527                                    ;*****************************************************************
1528                                    ;*TEST 7         RHCS2 - CONTROL AND STATUS 2
1529                                    ;*              TEST LOADING AND READING OF ALL POSSIBLE BITS IN THE HARDWARE
1530                                    ;*              REGISTERS.  USE A PETTERN OF WALKING 0'S (-2,-3,-5 ETC.) AND
1531                                    ;*              WALKING 1'S (1,2,4,10 ETC)
1532                                    ;*****************************************************************
1533  010072  000004                    TST7:   SCOPE
1534
1535  010074  012706  001000                    MOV     #STACK,SP       ;RESET STACK
1536  010100  012737  000007  002020            MOV     #TTNO,@#TSTNM   ;THIS SAVES TEST NUMBER
1537
1538  010106  013737  001630  010130            MOV     @#RHCS2,@#PRCS2+14      ;GET REGISTER ADDRESS
1539  010114  013777  001762  171506  PRCS2:    MOV     @#UNIT,@RHCS2   ;MOVE UNIT NO. UNDER TEST
1540  010122  004567  000046                    JSR     R5,BITST        ;TEST BITS IN REGISTER
1541  010126  020017                            .WORD   20017           ;ONLY THESE BITS ARE TESTED FOR READ/WRITE
1542  010130  176710                            .WORD   176710          ;ADDRESS OF REG. BEING TESTED
1543  010132  104001                            ERROR   1               ;INCORRECT DATA RECEIVED
1544  010134  000207                            RTS     PC              ;RETURN TO BLT3 ROUTINE
1545
1546
1547
1548
1549
1550
1551
1552                                    ;*****************************************************************
1553                                    ;*TEST 10        RHCS1 - CONTROL AND STATUS 1 REGISTER
1554                                    ;*              TEST LOADING AND READING OF ALL POSSIBLE BITS IN THE HARDWARE
1555                                    ;*              REGISTERS.  USE A PATTERN OF WALKING 0'S (-2,-3,-5 ETC) AND
1556                                    ;*              WALKING 1'S (1,2,4,10 ETC)
1557                                    ;*****************************************************************
1558  010136  000004                    TST10:  SCOPE
1559
1560  010140  012706  001000                    MOV     #STACK,SP       ;RESET STACK
1561  010144  012737  000010  002020            MOV     #TTNO,@#TSTNM   ;THIS SAVES TEST NUMBER
1562
1563  010152  013737  001632  010174            MOV     @#RHCS1,@#PRCS1+14      ;GET REGISTER ADDRESS
1564  010160  013777  001762  171442  PRCS1:    MOV     @#UNIT,@RHCS2   ;MOVE UNIT NO. UNDER TEST
1565  010166  004567  000002                    JSR     R5,BITST        ;TEST BITS IN REGISTER
1566  010172  001476                            .WORD   1476            ;ONLY THESE BITS ARE TESTED FOR READ/WRITE
1567  010174  176700                            .WORD   176700          ;ADDRESS OF REG. BEING TESTED
1568  010176  104001                            ERROR   1               ;INCORRECT DATA RECEIVED
1569  010200  000207                            RTS     PC              ;RETURN TO BLT3 ROUTINE
1570
1571
```

```
1572
1573
1574                                    ;*****************************************************************
1575                                    ;*TEST 11       RHCS1 - BIT # 13 - MCPE
1576                                    ;*      THIS FORCES A MASS BUS CONTROL PARITY ERROR
1577                                    ;*      BY SETTING 'PAT'  LOOKING FOR 'PAR', WRITING RHCS1
1578                                    ;*      AND READING RHER1
1579                                    ;*****************************************************************
1580  010202  000004                   TST11:  SCOPE
1581
1582  010204  012706  001000                   MOV     #STACK,SP       ;RESET STACK
1583  010210  012737  000011  002020           MOV     #TTNO,@#TSTNM   ;THIS SAVES TEST NUMBER
1584  010216  004737  040522                   JSR     PC,@#CLDISK     ;INIT AND SET UNIT NUMBER AND DEVICE -
1585                                                                   ;CPU REG. CORRESPONDENCE (R1-R4)
1586
1587                                           ;SET FORCED PARITY ERROR 'PAT'
1588  010222  052777  000020  171400           BIS     #PAT,@RHCS2     ;SET 'PAT' TO INVERT PARITY
1589                                                                   ;GENERATED
1590  010230  005077  171400                   CLR     @RHER1          ;WRITE DCL REGISTER USING BAD CONTROL PARITY
1591
1592                                           ;WITH THIS PARITY ERROR NOTHING WILL BE READ TILL IT IS
1593                                           ;CLEARED
1594
1595  010234  011137  001126                   MOV     @R1,@#$BDDAT    ;RHCS1 ---> $BDDAT
1596  010240  022737  104200  001126           CMP     #SC!DVA!RDY,@#$BDDAT ;COMPARE RHCS1 AFTER PARITY
1597                                                                   ;ERROR
1598  010246  001406                           BEQ     1$              ;BRANCH IF SC!DVA!RDY=1
1599  010250  012737  104200  001124           MOV     #SC!DVA!RDY,@#$GDDAT ;GOOD DATA
1600  010256  010137  040172                   MOV     R1,@#REGADR     ;REGISTER ADDRESS RHCS1
1601  010262  104001                           ERROR   1               ;SETTING PAT AND
1602                                                                   ;WRITING DCL REGISTER RHCS1
1603                                                                   ;DID NOT SET SC!DVA!RDY
1604                                                                   ;WITH 'PAT' BIT HIGH
1605  010264  013746  001762           1$:     MOV     @#UNIT,-(SP)    ;GET UNIT NUMBER
1606  010270  052716  000120                   BIS     #PAT!IR,(SP)    ;INCLUDE PAT AND IR
1607  010274  012637  001124                   MOV     (SP)+,@#$GDDAT  ;PUT ON STACK
1608  010300  011237  001126                   MOV     @R2,@#$BDDAT    ;RHCS2 ---> $BDDAT
1609  010304  023737  001124  001126           CMP     @#$GDDAT,@#$BDDAT ;COMPARE RHCS2
1610  010312  001403                           BEQ     2$              ;OK - SC!DVA!RDY ARE HIGH
1611  010314  010237  040172                   MOV     R2,@#REGADR     ;REGISTER ADDRESS
1612  010320  104001                           ERROR   1               ;READING DCL REGISTER RHCS2 DID NOT
1613                                                                   ;SHOW UNIT#!PAT!IR BITS HIGH
1614
1615  010322  011437  001126           2$:     MOV     @R4,@#$BDDAT    ;RHER1 ---> $BDDAT
1616  010326  022737  000010  001126           CMP     #PAR,@#$BDDAT   ;ERROR REGISTER RHER1 SHOULD
1617                                                                   ;HAVE 'PAR' SET
1618  010334  001406                           BEQ     3$              ;A - OK, IT DOES
1619  010336  012737  000010  001124           MOV     #PAR,@#$GDDAT   ;GOOD DATA
1620  010344  010437  040172                   MOV     R4,@#REGADR     ;FAILING REGISTER RHER1
1621  010350  104001                           ERROR   1               ;PARITY ERROR DID NOT
1622                                                                   ;SET 'PAR'
1623
1624  010352                           3$:
```

```
1625
1626
1627
1628                                    ;*****************************************************************
1629                                    ;*TEST 12        RHWC - WORD COUNT REGISTER
1630                                    ;*              TEST LOADING AND READING OF ALL POSSIBLE BITS IN THE HARDWARE
1631                                    ;*              REGISTERS.  USE A PATTERN OF WALKING 0'S (-2,-3,-5 ETC.) AND
1632                                    ;*              WALKING 1'S (1,2,4,10 ETC)
1633                                    ;*****************************************************************
1634  010352  000004                    TST12:  SCOPE
1635
1636  010354  012706  001000                    MOV     #STACK,SP       ;RESET STACK
1637  010360  012737  000012  002020            MOV     #TTNO,@#TSTNM   ;THIS SAVES TEST NUMBER
1638
1639  010366  013737  001624  010410            MOV     @#RHWC,@#PRWC+14        ;GET REGISTER ADDRESS
1640  010374  013777  001762  171226  PRWC:     MOV     @#UNIT,@RHCS2   ;MOVE UNIT NO. UNDER TEST
1641  010402  004567  027566                    JSR     R5,BITST        ;TEST BITS IN REGISTER
1642  010406  177777                            .WORD   177777          ;ONLY THESE BITS ARE TESTED FOR READ/WRITE
1643  010410  176702                            .WORD   176702          ;ADDRESS OF REG. BEING TESTED
1644  010412  104001                            ERROR   1               ;INCORRECT DATA RECEIVED
1645  010414  000207                            RTS     PC              ;RETURN TO BLT3 ROUTINE
1646
1647
1648
1649
1650
1651
1652                                    ;*****************************************************************
1653                                    ;*TEST 13        RHBA - UNIBUS ADDRESS REGISTER
1654                                    ;*              TEST LOADING AND READING OF ALL POSSIBLE BITS IN THE HARDWARE
1655                                    ;*              REGISTERS.  USE A PATTERN OF WALKING 0'S (-2,-3,-5 ETC.) AND
1656                                    ;*              WALKING 1'S (1,2,4,10 ETC)
1657                                    ;*****************************************************************
1658  010416  000004                    TST13:  SCOPE
1659
1660  010420  012706  001000                    MOV     #STACK,SP       ;RESET STACK
1661  010424  012737  000013  002020            MOV     #TTNO,@#TSTNM   ;THIS SAVES TEST NUMBER
1662
1663  010432  013737  001626  010454            MOV     @#RHBA,@#PRBA+14        ;GET REGISTER ADDRESS
1664  010440  013777  001762  171162  PRBA:     MOV     @#UNIT,@RHCS2   ;MOVE UNIT NO. UNDER TEST
1665  010446  004567  027522                    JSR     R5,BITST        ;TEST BITS IN REGISTER
1666  010452  177776                            .WORD   177776          ;ONLY THESE BITS ARE TESTED FOR READ/WRITE
1667  010454  176704                            .WORD   176704          ;ADDRESS OF REG. BEING TESTED
1668  010456  104001                            ERROR   1               ;INCORRECT DATA RECEIVED
1669  010460  000207                            RTS     PC              ;RETURN TO BLT3 ROUTINE
1670
1671
1672
```

```
1673
1674
1675
1676                                    ;*****************************************************************
1677                                    ;*TEST 14        RHER1 - ERROR REGISTER #1
1678                                    ;*              TEST LOADING AND READING OF ALL POSSIBLE BITS IN THE HARDWARE
1679                                    ;*              REGISTERS.  USE A PATTERN OF WALKING 0'S (-2,-3,-5 ETC.) AND
1680                                    ;*              WALKING 1'S (1,2,4,10 ETC)
1681                                    ;*****************************************************************
1682  010462  000004                   TST14:  SCOPE
1683
1684  010464  012706  001000                    MOV     #STACK,SP       ;RESET STACK
1685  010470  012737  000014  002020           MOV     #TTNO,@#TSTNM   ;THIS SAVES TEST NUMBER
1686
1687  010476  013737  001634  010520           MOV     @#RHER1,@#PRER1+14      ;GET REGISTER ADDRESS
1688  010504  013777  001762  171116  PRER1:   MOV     @#UNIT,@RHCS2   ;MOVE UNIT NO. UNDER TEST
1689  010512  004567  027456                   JSR     R5,BITST        ;TEST BITS IN REGISTER
1690  010516  177777                           .WORD   177777          ;ONLY THESE BITS ARE TESTED FOR READ/WRITE
1691  010520  176714                           .WORD   176714          ;ADDRESS OF REG. BEING TESTED
1692  010522  104001                           ERROR   1               ;INCORRECT DATA RECEIVED
1693  010524  000207                           RTS     PC              ;RETURN TO BLT3 ROUTINE
1694
1695
```

```
                                        ;*****************************************************************
                                        ;*TEST 15       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)
                                        ;*****************************************************************
010526  000004                  TST15:  SCOPE

010530  012737  000015  002020          MOV     #TTNO,@#TSTNM   ;THIS SAVES TEST NUMBER
010536  004737  040522                  JSR     PC,@#CLDISK     ;SET UNIT NUMBER AND INIT
010542  013700  001652                  MOV     @#RHMR, R0      ;R0 HAS MAINTENANCE REG. ADR.
010546  012701  000001                  MOV     #1,R1           ;R1 HAS DATA
010552  012702  000005                  MOV     #5,R2           ;R2 HAS COUNT OF NUMBER OF BITS
010556  012710  000001          1$:     MOV     #DMD,@R0        ;SET DIAGNOSTIC MODE BIT
010562  050110                  BIS     R1,@R0          ;SET DATA IN RHMR
010564  010146                  MOV     R1,-(SP)        ;SAVE DATA FOR COMPARES
010566  052716  000401                  BIS     #DMD!400,(SP)   ;INCLUDE BIT 0
010572  011637  001124                  MOV     (SP),@#$GDDAT   ;SAVE FOR ERROR PRINTOUT
010576  022610                  CMP     (SP)+,@R0       ;COMPARE DATA
010600  001405                  BEQ     2$              ;BRANCH IF GOOD
010602  011067  170320                  MOV     @R0,$BDDAT      ;BAD DATA
010606  010037  040172                  MOV     R0,@#REGADR     ;FAILING REG. ADR.
010612  104001                  ERROR   1               ;MAINTENANCE REGISTER
                                                        ;FAILED TO SET INDICATED
                                                        ;BITS
010614  000241          2$:     CLC                     ;CLEAR CARRY
010616  006101                  ROL     R1              ;GET NEXT DATA
010620  052701  000400                  BIS     #400,R1         ;SET UNUSED BITS
010624  042701  001000                  BIC     #BIT09,R1       ;CLEAR READ ONLY BIT
010630  005302                  DEC     R2              ;COUNT
010632  001351                  BNE     1$              ;BRANCH IF 5 BITS NOT DONE


                                        ;NOW FLOAT A 0

010634  012701  000435                  MOV     #435,R1         ;R1 HAS DATA
010640  012702  000005                  MOV     #5,R2           ;R2 HAS COUNT BITS
010644  012710  000001          3$:     MOV     #DMD,@R0        ;SET DIAGNOSTIC MODE BITS
010650  050110                  BIS     R1,@R0          ;SET DATA IN RHMR
010652  020110                  CMP     R1,@R0          ;COMPARE DATA
010654  001407                  BEQ     4$              ;BRANCH IF GOOD
010656  010137  001124                  MOV     R1,@#$GDDAT     ;GOOD DATA
010662  011037  001126                  MOV     @R0,@#$BDDAT    ;BAD DATA
010666  010037  040172                  MOV     R0,@#REGADR     ;FAILING REG. ADR. RHMR
010672  104001                  ERROR   1               ;MAINTENANCE REGISTER
                                                        ;DOES NOT ALLOW WRITING
                                                        ;ZEROS
010674  000261          4$:     SEC                     ;SET CARRY
010676  006101                  ROL     R1              ;GET NEXT DATA
010700  042701  001340                  BIC     #BIT05!BIT06!BIT07!BIT09,R1   ;CLEAR READ ONLY BIT
010704  052701  000400                  BIS     #BIT08,R1       ;SET BIT ZEROED BY ROL
```

```
1750  010710  005302                          DEC     R2              ;COUNT IF 5 BITS DONE
1751  010712  001354                          BNE     3$              ;BRANCH IF INCOMPLETE
1752
1753
```

```
1754
1755
1756
1757                                    ;*************************************************************
1758                                    ;*TEST 16        RHDST - DESIRED SECTOR/TRACK ADDRESS
1759                                    ;*              TEST LOADING AND READING OF ALL POSSIBLE BITS IN THE HARDWARE
1760                                    ;*              REGISTERS.  USE A PATTERN OF WALKING 0'S (-2,-3,-5 ETC.) AND
1761                                    ;*              WALKING 1'S (1,2,4,10 ETC)
1762                                    ;*************************************************************
1763  010714  000004                    TST16:  SCOPE
1764
1765
1766  010716  012706  001000                    MOV     #STACK,SP       ;RESET STACK
1767  010722  012737  000016  002020            MOV     #TTNO,@#TSTNM   ;THIS SAVES TEST NUMBER
1768
1769  010730  013737  001636  010752            MOV     @#RHDST,@#PRDST+14      ;GET REGISTER ADDRESS
1770  010736  013777  001762  170664  PRDST:    MOV     @#UNIT,@RHCS2   ;MOVE UNIT NO. UNDER TEST
1771  010744  004567  027224                    JSR     R5,BITST        ;TEST BITS IN REGISTER
1772  010750  017437                            .WORD   17437           ;ONLY THESE BITS ARE TESTED FOR READ/WRITE
1773  010752  176706                            .WORD   176706          ;ADDRESS OF REG. BEING TESTED
1774  010754  104001                            ERROR   1               ;INCORRECT DATA RECEIVED
1775  010756  000207                            RTS     PC              ;RETURN TO BLT3 ROUTINE
1776
1777
1778
1779
1780
1781
1782
1783                                    ;*************************************************************
1784                                    ;*TEST 17        RHER2 - ERROR REGISTER #2
1785                                    ;*              TEST LOADING AND READING OF ALL POSSIBLE BITS IN THE HARDWARE
1786                                    ;*              REGISTERS.  USE A PATTERN OF WALKING 0'S (-2,-3,-5 ETC.) AND
1787                                    ;*              WALKING 1'S (1,2,4,10 ETC)
1788                                    ;*************************************************************
1789  010760  000004                    TST17:  SCOPE
1790
1791
1792  010762  012706  001000                    MOV     #STACK,SP       ;RESET STACK
1793  010766  012737  000017  002020            MOV     #TTNO,@#TSTNM   ;THIS SAVES TEST NUMBER
1794
1795  010774  013737  001640  011016            MOV     @#RHER2,@#PRER2+14      ;GET REGISTER ADDRESS
1796  011002  013777  001762  170620  PRER2:    MOV     @#UNIT,@RHCS2   ;MOVE UNIT NO. UNDER TEST
1797  011010  004567  027160                    JSR     R5,BITST        ;TEST BITS IN REGISTER
1798  011014  177777                            .WORD   177777          ;ONLY THESE BITS ARE TESTED FOR READ/WRITE
1799  011016  176740                            .WORD   176740          ;ADDRESS OF REG. BEING TESTED
1800  011020  104001                            ERROR   1               ;INCORRECT DATA RECEIVED
1801  011022  000207                            RTS     PC              ;RETURN TO BLT3 ROUTINE
1802
1803
1804
1805
```

```
                                ;*****************************************************************
                                ;*TEST 20        RHOF - MARGIN/OFFSET 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)
                                ;*****************************************************************
011024  000004          TST20:  SCOPE

011026  012706  001000          MOV     #STACK,SP       ;RESET STACK
011032  012737  000020  002020  MOV     #TTNO,@#TSTNM   ;THIS SAVES TEST NUMBER

011040  013737  001642  011062  MOV     @#RHOF,@#PROF+14        ;GET REGISTER ADDRESS
011046  013777  001762  171554  PROF:   MOV     @#UNIT,@RHCS2   ;MOVE UNIT NO. UNDER TEST
011054  004567  027114          JSR     R5,BITST        ;TEST BITS IN REGISTER
011060  016277                  .WORD   16277           ;ONLY THESE BITS ARE TESTED FOR READ/WRITE
011062  176732                  .WORD   176732          ;ADDRESS OF REG. BEING TESTED
011064  104001                  ERROR   1               ;INCORRECT DATA RECEIVED
011066  000207                  RTS     PC              ;RETURN TO BLT3 ROUTINE



                                ;*****************************************************************
                                ;*TEST 21        RHCA - DESIRED CYLINDER 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)
                                ;*****************************************************************
011070  000004          TST21:  SCOPE

011072  012706  001000          MOV     #STACK,SP       ;RESET STACK
011076  012737  000021  002020  MOV     #TTNO,@#TSTNM   ;THIS SAVES TEST NUMBER

011104  013737  001644  011126  MOV     @#RHCA,@#PRCA+14        ;GET REGISTER ADDRESS
011112  013777  001762  170510  PRCA:   MOV     @#UNIT,@RHCS2   ;MOVE UNIT NO. UNDER TEST
011120  004567  027050          JSR     R5,BITST        ;TEST BITS IN REGISTER
011124  001777                  .WORD   1777            ;ONLY THESE BITS ARE TESTED FOR READ/WRITE
011126  176734                  .WORD   176734          ;ADDRESS OF REG. BEING TESTED
011130  104001                  ERROR   1               ;INCORRECT DATA RECEIVED
011132  000207                  RTS     PC              ;RETURN TO BLT3 ROUTINE
```

```
                                        ;*****************************************************************
                                        ;*TEST 22       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)
                                        ;*****************************************************************
011134  000004                  TST22:  SCOPE

011136  012706  001000                  MOV     #STACK,SP       ;RESET STACK
011142  012737  000022  002020          MOV     #TTNO,@#TSTNM   ;THIS SAVES TEST NUMBER

011150  013737  001646  011172          MOV     @#RHER3,@#PRER3+14      ;GET REGISTER ADDRESS
011156  013777  001762  170444  PRER3:  MOV     @#UNIT,@RHCS2   ;MOVE UNIT NO. UNDER TEST
011164  004567  027004                  JSR     R5,BITST        ;TEST BITS IN REGISTER
011170  177777                          .WORD   177777          ;ONLY THESE BITS ARE TESTED FOR READ/WRITE
011172  176742                          .WORD   176742          ;ADDRESS OF REG. BEING TESTED
011174  104001                          ERROR   1               ;INCORRECT DATA RECEIVED
011176  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 23       CONTROL AND STATUS 2 (RHCS 2) - 'NED'
                                ;*      THIS TESTS THE UNIT SELECT BITS #0-2 (US1-4)
                                ;*      AND NON-EXISTENT DRIVE BIT #12 (NED)
                                ;*
                                ;*      THE OTHER RHCS2 BITS ARE NOT TESTED HERE
                                ;*****************************************************************
011200  000004                  TST23:  SCOPE
011202  012706  001000                  MOV     #STACK,SP       ;RESET STACK
011206  012737  000023  002020          MOV     #TTNO,@#TSTNM   ;THIS SAVES TEST NUMBER
011214  004737  040522                  JSR     PC,@#CLDISK     ;HERE IT IS USED TO SETUP HARDWARE/
                                                                ;CPU REGISTER CORRESPONDENCE
                                                                ;R1=RHCS1
                                                                ;R2=RHCS2
                                                                ;R3=RHDS1
                                                                ;R4=RHER1
011220  005037  001774                  CLR     @#ERFLGS        ;CLEAR ERROR FLAG

                                ;SIMULATED DISK AREA WILL BE USED AS A TEMPORARY
                                ;STORAGE TABLE FOR DRIVES PRESENT

011224  012701  047070                  MOV     #DISK,R1        ;LOAD TABLE POINTER
011230  012700  000010                  MOV     #8.,R0          ;LOAD TABLE LOCATION COUNTER
011234  012721  177777          1$:     MOV     #-1,(R1)+       ;FILL 8 LOCATIONS WITH -1
011240  005300                          DEC     R0              ;COUNT DOWN ONE LOCATION
011242  001374                          BNE     1$              ;BRANCH IF 8 NOT DONE

011244  005012                          CLR     @R2             ;SELECT UNIT NO.0 (U2!U1!U0=0)
011246  012700  000010                  MOV     #8.,R0          ;RELOAD TABLE LOCATION COUNTER
011252  012701  047070                  MOV     #DISK,R1        ;RELOAD THE TABLE POINTER
011256  005714                  2$:     TST     @R4             ;READ A DRIVE REGISTER (RHER1)
011260  032712  010000                  BIT     #NED,@R2        ;NON EXISTENT DRIVE BIT = 0 ?
011264  001415                          BEQ     3$              ;YES...DRIVE PRESENT, CHECK THE TYPE
011266  005300                  7$:     DEC     R0              ;NO...DECREMENT DRIVE COUNT
011270  001434                          BEQ     4$              ;CHECK RESULTS IF 8 DRIVES DONE

011272  011246                  10$:    MOV     @R2,-(SP)       ;PUT 'RHCS2' ON THE STACK
011274  042716  177770                  BIC     #^C7,(SP)       ;MASK ALL BUT THE UNIT NUMBER
011300  005216                          INC     (SP)            ;INCREMENT THE UNIT NUMBER
011302  013703  001632                  MOV     @#RHCS1,R3      ;GET RHCS1 ADDRESS
011306  005203                          INC     R3              ;ADDRESS UPPER BYTE OF RHCS1
011310  112713  000100                  MOVB    #100,@R3        ;SET TRE IN RHCS1
                                                                ;WITHOUT ADDRESSING DRIVE
011314  012612                          MOV     (SP)+,@R2       ;RHCS2 HAS INCREMENTED UNIT WITH NED CLEARED
011316  000757                          BR      2$              ;TEST FOR NEXT DRIVE

011320  022777  024020  170330  3$:     CMP     #24020,@RHDT    ;IS THIS A DUAL PORT RP04 ?
011326  001405                          BEQ     8$              ;TYPE IT OUT IF SO
011330  022777  020020  170320          CMP     #20020,@RHDT    ;IS THIS A SINGLE PORT RP04 ?
011336  001401                          BEQ     8$              ;TYPE IT OUT IF SO
```

```
1941  011340  000752                          BR      7$              ;NO RP04 FOUND SO CHECK NEXT UNIT
1942
1943  011342  012746  000010          8$:     MOV     #8.,-(SP)       ;LOAD MAX NO. OF DRIVES
1944  011346  160016                          SUB     R0,(SP)         ;(SP) NOW HAS THE PRESENT DRIVE NO.
1945  011350  012621                          MOV     (SP)+,(R1)+     ;LOAD TABLE, INCR TABLE LOCATION &
1946                                                                  ;RESTORE THE STACK TO WHERE IT WAS
1947  011352  005300                          DEC     R0              ;DECREMENT THE DRIVE COUNT
1948  011354  001402                          BEQ     4$              ;CHECK RESULTS IF 8 UNITS CHECKED
1949  011356  005212                          INC     @R2             ;SELECT NEXT UNIT
1950  011360  000736                          BR      2$              ;GO TEST IT
1951
1952  011362  004037  041416          4$:     JSR     R0,@#COMPAR     ;COMPARE RESULTS
1953  011366  001742                          UNITS                   ;RHER1/RHAS DERIVED DATA
1954  011370  047070                          DISK                    ;'NED' TEST DATA
1955  011372  000010                          8.                      ;NO.OF WORDS TO COMPARE
1956  011374  011402                          5$                      ;RETURN FOR ERROR HEADER
1957  011376  011430                          6$                      ;RETURN FOR ERROR DATA
1958  011400  011546                          13$                     ;RETURN FOR GOOD COMPARISON (NEXT TEST)
1959
1960                                          ;SPECIAL 'NED'/'RHAS' TABLE TYPE OUT ROUTINE
1961
1962  011402  104022          5$:     ERROR   22
1963  011404  012703  000010                MOV     #8.,R3          ;LENGTH OF BOTH UNIT TABLES
1964  011410  012701  001742                MOV     #UNITS,R1       ;ADDRESS OF RHAS/RHER1 UNITS TABLE
1965  011414  012702  047070                MOV     #DISK,R2        ;ADDRESS OF 'NED' RHCS2 UNITS TABLE
1966  011420  012137  001124          14$:    MOV     (R1)+,@#$GDDAT  ;LOAD RHAS UNIT NO.INTO "$GDDAT" AND
1967                                                                  ;INCREMENT THE TABLE LOCATION
1968  011424  012237  001126                MOV     (R2)+,@#$BDDAT  ;LOAD 'NED' UNIT NO. INTO "$BDDAT"
1969                                                                  ;& INCR TABLE LOCATION
1970
1971  011430  032737  020000  177570  6$:     BIT     #SW13,@#SWR     ;INHIBIT ERROR TYPE OUTS ?
1972  011436  001043                          BNE     13$             ;YES...EXIT
1973  011440  022737  177777  001124          CMP     #-1,@#$GDDAT    ;DOES RHAS UNIT TABLE LOCATION = -1 ?
1974  011446  001413                          BEQ     11$             ;YES...DON'T TYPE IT - CHECK 'NED' TABLE
1975  011450  104400  057450                TYPE    ,SPACE8         ;NO...TAB OVER PC COLUMN
1976  011454  104400  057450                TYPE    ,SPACE8         ;TAB OVER THE TEST NO. COLUMN
1977  011460  013746  001124                MOV     @#$GDDAT,-(SP)  ;;SAVE @#$GDDAT FOR TYPEOUT
1978  011464  104404                        TYPOS                   ;;GO TYPE--OCTAL ASCII
1979  011466     006                        .BYTE   6               ;;TYPE 6 DIGITS
1980  011467     000                        .BYTE   0               ;;SUPPRESS LEADING ZEROS
1981  011470  104400  057456                TYPE    ,SPACE2         ;SPACE OVER TO THE NEXT COLUMN
1982  011474  000406                        BR      12$             ;CHECK THE 'NED' UNIT TABLE
1983
1984  011476  104400  057450          11$:    TYPE    ,SPACE8         ;TAB OVER THE PC COMUMN
1985  011502  104400  057450                TYPE    ,SPACE8         ;TAB OVER THE TEST NO. COLUMN
1986  011506  104400  057450                TYPE    ,SPACE8         ;TAB OVER THE RHAS UNIT COLUMN
1987
1988  011512  022737  177777  001126  12$:     CMP     #-1,@#$BDDAT    ;DOES 'NED' UNIT TABLE LOCATION = - 1 ?
1989  011520  001404                          BEQ     9$              ;YES...DON'T TYPE IT
1990  011522  013746  001126                MOV     @#$BDDAT,-(SP)  ;;SAVE @#$BDDAT FOR TYPEOUT
1991  011526  104404                        TYPOS                   ;;GO TYPE--OCTAL ASCII
1992  011530     006                        .BYTE   6               ;;TYPE 6 DIGITS
1993  011531     000                        .BYTE   0               ;;SUPPRESS LEADING ZEROS
1994
```

```
1995  011532  104400  001215          9$:     TYPE    $CRLF           ;FOR THE NEXT LINE IN BOTH TABLES
1996  011536  005303                          DEC     R3              ;COUNT DOWN 2 TABLES LOCATION COUNTER
1997  011540  001327                          BNE     14$             ;IF NOT = 0 TYPE OUT NEXT 2 LOCATIONS
1998  011542  062706  000014                  ADD     #14,SP          ;ADJUST STACK FOR NO "POP" & RTS FROM "COMPAR"
1999
2000  011546                          13$:
2001
2002
2003
2004

        ;/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:
        ;/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:
        ;/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:/*\:


        ;IN THE ABOVE TEST BITS 0,1,2, AND BIT 12 ARE TESTED

        ;IF THE "DRIVES PRESENT" TYPE OUT DOES NOT AGREE WITH WHAT WAS
        ;FOUND USING RHER1 & RHAS, THEN THE ERROR IS IN THE LOGIC
        ;FOR BIT12(NED), OR UNIT SELECT(BIT 0 TO 2), OR RHER1, OR RHAS

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


        ;/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*
        ;/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*
        ;/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*/:\*
```

```
                                ;*****************************************************************
                                ;*TEST 24       CONTROL AND STATUS 2 (RHCS2) - 'CLR'
                                ;THIS TESTS THE UNIT SELECT BITS (US1-4) AND CLEAR BIT #5 (CLR)

                                ;ALL REGISTERS ARE LOADED WITH ALL ONES EXCEPT BIT #0 AND #6
                                ;WHICH ARE "GO" AND "INTERRUPT ENABLE", THEN 'CLR' IS GIVEN
                                ;(RHDB IS READ FIRST AS THIS WILL SET DTL IN RHCS2 AND
                                ;SC AND TRE IN RHCS1.)

                                ;ANOTHER CLR IS GIVEN THEN ALL OTHER REGISTERS ARE READ
                                ;*****************************************************************
011546  000004                  TST24:  SCOPE

011550  004737  040522                  JSR     PC,@#CLDISK     ;SET REGISTERS AND CLEAR
011554  005037  001774                  CLR     @#ERFLG$        ;CLEAR ANY ERRORS

                                        ;FILL ALL POSSIBLE BITS WITH ONES

011560  012777  177777  170034          MOV     #177777,@RHDB   ;BUS ADDRESS REGISTER GETS 177777
011566  012777  177777  170030          MOV     #177777,@RHWC   ;WORD COUNT REGISTER GETS 177777
011574  012777  177777  170024          MOV     #177777,@RHBA   ;BUS ADDRESS REGISTER GETS 177777
011602  052777  157010  170020          BIS     #157010,@RHCS2  ;CONTROL AND STATUS 2 GETS 157010
011610  012777  001476  170014          MOV     #1476,@RHCS1    ;CONTROL AND STATUS REGISTER/GETS 1476
011616  012777  177777  170010          MOV     #177777,@RHER1  ;ERROR REGISTER1 GETS 177777
011624  012777  017437  170004          MOV     #17437,@RHDST   ;DESIRED SECTOR TRACK
011632  012777  177777  170000          MOV     #177777,@RHER2  ;ERROR REGISTER 2
011640  012777  016277  167774          MOV     #16277,@RHOF    ;OFFSET REGISTER
011646  012777  177777  167770          MOV     #177777,@RHCA   ;DESIRED CYLINDER
011654  012777  177777  167764          MOV     #177777,@RHER3  ;ERROR REGISTER 3
011662  012777  000001  167762          MOV     #DMD,@RHMR      ;MAINTENANCE REGISTER
011670  012777  177777  167754          MOV     #177777,@RHMR   ;MAINTENANCE REGISTER

011676  052712  000040                  BIS     #CLR,   @R2     ;CLEAR ALL POSSIBLE BITS
011702  013712  001762                  MOV     @#UNIT,@R2      ;REINSTATE UNIT NO.
011706  012700  001622                  MOV     #RHDB,  R0      ;R0 CONTAINS ADDR. OF ADDR. OF REG.

                                        ;DATA BUFFER REGISTER

011712  012737  177777  001124          MOV     #177777,@#$GDDAT;GOOD DATA FOR ERROR
011720  011037  040172                  MOV     @R0,@#REGADR    ;REGISTER ADDRESS
011724  013037  001126                  MOV     @(R0)+,@#$BDDAT ;TEST DATA
011730  023737  001124  001126          CMP     @#$GDDAT,@#$BDDAT;COMPARE GOOD WITH TEST DATA
011736  001401                          BEQ     2$              ;BRANCH IF GOOD
011740  104001                          ERROR   1               ;RHDB DID NOT HAVE ALL ONES
                                                                ;AFTER A CLR IN RHCS2

011742  052712  000040          2$:     BIS     #CLR,@R2        ;SET CLEAR AGAIN BECAUSE
                                                                ;READING RHDB AFTER CLEARING  WILL
                                                                ;SET DLT SC AND TRE
011746  013712  001762                  MOV     @#UNIT,@R2      ;REINSTATE UNIT NO.
```

```
2088
2089                                        ;WORD COUNT REGISTER
2090
2091
2092  011752  012737  177777  001124  3$:   MOV   #177777,       @#SGDDAT ;GOOD DATA FOR ERROR TYPEOUT
2093  011760  013037  001126              MOV   @(R0)+,@#SBDDAT ;TEST DATA
2094  011764  022737  177777  001126      CMP   #177777,       @#SBDDAT;COMPARE DATA
2095  011772  001402                      BEQ   4$              ;BRANCH IF GOOD
2096  011774  004737  012560              JSR   PC,@#ERCS2C     ;JUMP TO ERROR FOR CLR (BIT 5)
2097                                                            ;IN RHCS2
2098
2099                                        ;BUS ADDRESS REGISTER
2100
2101
2102  012000  012737  000000  001124  4$:   MOV   #0,     @#SGDDAT ;GOOD DATA FOR ERROR TYPEOUT
2103  012006  013037  001126              MOV   @(R0)+,@#SBDDAT ;TEST DATA
2104  012012  022737  000000  001126      CMP   #0,     @#SBDDAT;COMPARE DATA
2105  012020  001402                      BEQ   5$              ;BRANCH IF GOOD
2106  012022  004737  012560              JSR   PC,@#ERCS2C     ;JUMP TO ERROR FOR CLR (BIT 5)
2107                                                            ;IN RHCS2
2108
2109                                        ;CONTROL AND STATUS 2 REGISTER
2110
2111
2112  012026  012746  000100          5$:   MOV   #100,    -(SP)   ;INCLUDE IR
2113  012032  053716  001762              BIS   @#UNIT, (SP)    ;SET UNIT NO.
2114  012036  012637  001124              MOV   (SP)+,  @#SGDDAT;GOOD DATA FOR TYPE OUT
2115  012042  013037  001126              MOV   @(R0)+,@#SBDDAT ;TEST DATA
2116  012046  023737  001124  001126      CMP   @#SGDDAT,@#SBDDAT;COMPARE DATA
2117  012054  001402                      BEQ   6$              ;BRANCH IF GOOD
2118  012056  004737  012560              JSR   PC,@#ERCS2C     ;JUMP TO ERROR FOR CLR (BIT 5)
2119                                                            ;IN RHCS2
2120
2121
2122                                        ;CONTROL AND STATUS 1 REGISTER
2123
2124
2125  012062  012737  004276  001124  6$:   MOV   #4276,  @#SGDDAT ;GOOD DATA FOR ERROR TYPEOUT
2126  012070  013037  001126              MOV   @(R0)+,@#SBDDAT ;TEST DATA
2127  012074  022737  004276  001126      CMP   #4276,  @#SBDDAT;COMPARE DATA
2128  012102  001402                      BEQ   7$              ;BRANCH IF GOOD
2129  012104  004737  012560              JSR   PC,@#ERCS2C     ;JUMP TO ERROR FOR CLR (BIT 5)
2130                                                            ;IN RHCS2
2131
2132                                        ;ERROR 1 REGISTER
2133
2134
2135  012110  012737  000000  001124  7$:   MOV   #0,     @#SGDDAT ;GOOD DATA FOR ERROR TYPEOUT
2136  012116  013037  001126              MOV   @(R0)+,@#SBDDAT ;TEST DATA
2137  012122  022737  000000  001126      CMP   #0,     @#SBDDAT;COMPARE DATA
2138  012130  001402                      BEQ   10$             ;BRANCH IF GOOD
2139  012132  004737  012560              JSR   PC,@#ERCS2C     ;JUMP TO ERROR FOR CLR (BIT 5)
2140                                                            ;IN RHCS2
2141
```

```
2142                                            ;DESIRED SECTOR/TRACK REGISTER
2143
2144
2145  012136  012737  017437  001124  10$:     MOV     #17437, @#$GDDAT ;GOOD DATA FOR ERROR TYPEOUT
2146  012144  013037  001126                  MOV     @(R0)+,@#$BDDAT ;TEST DATA
2147  012150  022737  017437  001126          CMP     #17437, @#$BDDAT;COMPARE DATA
2148  012156  001402                          BEQ     11$             ;BRANCH IF GOOD
2149  012160  004737  012560                  JSR     PC,@#ERCS2C     ;JUMP TO ERROR FOR CLR (BIT 5)
2150                                                                  ;IN RHCS2
2151
2152                                            ;ERROR 2 REGISTER
2153
2154
2155  012164  012737  000000  001124  11$:     MOV     #0,     @#$GDDAT ;GOOD DATA FOR ERROR TYPEOUT
2156  012172  013037  001126                  MOV     @(R0)+,@#$BDDAT ;TEST DATA
2157  012176  022737  000000  001126          CMP     #0,     @#$BDDAT;COMPARE DATA
2158  012204  001402                          BEQ     12$             ;BRANCH IF GOOD
2159  012206  004737  012560                  JSR     PC,@#ERCS2C     ;JUMP TO ERROR FOR CLR (BIT 5)
2160                                                                  ;IN RHCS2
2161
2162                                            ;OFFSET REGISTER
2163
2164
2165  012212  012737  116000  001124  12$:     MOV     #116000,        @#$GDDAT ;GOOD DATA FOR ERROR TYPEOUT
2166  012220  013037  001126                  MOV     @(R0)+,@#$BDDAT ;TEST DATA
2167  012224  022737  116000  001126          CMP     #116000,        @#$BDDAT;COMPARE DATA
2168  012232  001402                          BEQ     13$             ;BRANCH IF GOOD
2169  012234  004737  012560                  JSR     PC,@#ERCS2C     ;JUMP TO ERROR FOR CLR (BIT 5)
2170                                                                  ;IN RHCS2
2171
2172                                            ;DESIRED CYLINDER ADDRESS REGISTER
2173
2174
2175  012240  012737  001777  001124  13$:     MOV     #1777,  @#$GDDAT ;GOOD DATA FOR ERROR TYPEOUT
2176  012246  013037  001126                  MOV     @(R0)+,@#$BDDAT ;TEST DATA
2177  012252  022737  001777  001126          CMP     #1777,  @#$BDDAT;COMPARE DATA
2178  012260  001402                          BEQ     14$             ;BRANCH IF GOOD
2179  012262  004737  012560                  JSR     PC,@#ERCS2C     ;JUMP TO ERROR FOR CLR (BIT 5)
2180                                                                  ;IN RHCS2
2181
2182                                            ;ERROR 3 REGISTER
2183
2184
2185  012266  012737  000000  001124  14$:     MOV     #0,     @#$GDDAT ;GOOD DATA FOR ERROR TYPEOUT
2186  012274  013037  001126                  MOV     @(R0)+,@#$BDDAT ;TEST DATA
2187  012300  022737  000000  001126          CMP     #0,     @#$BDDAT;COMPARE DATA
2188  012306  001402                          BEQ     15$             ;BRANCH IF GOOD
2189  012310  004737  012560                  JSR     PC,@#ERCS2C     ;JUMP TO ERROR FOR CLR (BIT 5)
2190                                                                  ;IN RHCS2
2191
2192                                            ;ATTENTION SUMMARY REGISTER
2193
2194  012314  013037  001126          15$:     MOV     @(R0)+,@#$BDDAT         ;GET RHAS CONTENTS
2195  012320  012737  000000  001124          MOV     #0,@#$GDDAT             ;GOOD DATA FOR ERROR TYPE OUT
```

```
2196  012326  123737  001124  001126          CMPB    @#$GDDAT,@#$BDDAT       ;COMPARE FOR RHAS
2197  012334  001402                          BEQ     16$                     ;BRANCH IF GOOD
2198  012336  004737  012560                  JSR     PC,@#ERCS2C             ;JUMP TO ERROR FOR CLR (BIT 5)
2199                                                                          ;IN RHCS2
2200
2201                                          ;MAINTAINABILITY REGISTER
2202
2203
2204  012342  012737  000400  001124  16$:    MOV     #400,   @#$GDDAT ;GOOD DATA FOR ERROR TYPEOUT
2205  012350  013037  001126                  MOV     @(R0)+,@#$BDDAT ;TEST DATA
2206  012354  022737  000400  001126          CMP     #400,   @#$BDDAT;COMPARE DATA
2207  012362  001402                          BEQ     17$             ;BRANCH IF GOOD
2208  012364  004737  012560                  JSR     PC,@#ERCS2C     ;JUMP TO ERROR FOR CLR (BIT 5)
2209                                                                  ;IN RHCS2
2210
2211                                          ;DRIVE STATUS REGISTER
2212
2213  012370  012737  000600  001124  17$:    MOV     #600,@#$GDDAT   ;GOOD DATA FOR ERROR TYPEOUT
2214  012376  013046                          MOV     @(R0)+,-(SP)    ;GET RHDS1
2215  012400  011637  001126                  MOV     (SP),@#$BDDAT   ;TEST DATA
2216  012404  042716  001100                  BIC     #VV!PROG,(SP)   ;CLEAR VV AND PROG
2217  012410  022726  000600                  CMP     #600,(SP)+      ;COMPARE DATA
2218  012414  001402                          BEQ     20$             ;BRANCH IF GOOD
2219  012416  004737  012560                  JSR     PC,@#ERCS2C     ;JUMP TO ERROR FOR CLR (BIT 5)
2220                                                                  ;IN RHCS2
2221
2222                                          ;DRIVE TYPE
2223
2224
2225  012422  013737  001776  001124  20$:    MOV     @#SAVDT,        @#$GDDAT ;GOOD DATA FOR ERROR TYPEOUT
2226  012430  013037  001126                  MOV     @(R0)+,@#$BDDAT ;TEST DATA
2227  012434  023737  001776  001126          CMP     @#SAVDT,        @#$BDDAT;COMPARE DATA
2228  012442  001402                          BEQ     21$             ;BRANCH IF GOOD
2229  012444  004737  012560                  JSR     PC,@#ERCS2C     ;JUMP TO ERROR FOR CLR (BIT 5)
2230                                                                  ;IN RHCS2
2231
2232
2233                                          ;SERIAL NUMBER REGISTER
2234
2235
2236
2237  012450  013737  002000  001124  21$:    MOV     @#SAVSN,        @#$GDDAT ;GOOD DATA FOR ERROR TYPEOUT
2238  012456  013037  001126                  MOV     @(R0)+,@#$BDDAT ;TEST DATA
2239  012462  023737  002000  001126          CMP     @#SAVSN,        @#$BDDAT;COMPARE DATA
2240  012470  001402                          BEQ     22$             ;BRANCH IF GOOD
2241  012472  004737  012560                  JSR     PC,@#ERCS2C     ;JUMP TO ERROR FOR CLR (BIT 5)
2242                                                                  ;IN RHCS2
2243
2244
2245                                          ;ECC1 POSITION
2246
2247
2248  012476  012737  000000  001124  22$:    MOV     #0,     @#$GDDAT ;GOOD DATA FOR ERROR TYPEOUT
2249  012504  013037  001126                  MOV     @(R0)+,@#$BDDAT ;TEST DATA
```

```
2250  012510  022737  000000  001126          CMP     #0,     @#SBDDAT;COMPARE DATA
2251  012516  001402                          BEQ     23$             ;BRANCH IF GOOD
2252  012520  004737  012560                  JSR     PC,@#ERCS2C     ;JUMP TO ERROR FOR CLR (BIT 5)
2253                                                                  ;IN RHCS2
2254
2255
2256                                          ;ECC2 PATTERN
2257
2258
2259  012524  012737  000000  001124  23$:    MOV     #0,     @#SGDDAT ;GOOD DATA FOR ERROR TYPEOUT
2260  012532  013037  001126                  MOV     @(R0)+,@#SBDDAT ;TEST DATA
2261  012536  022737  000000  001126          CMP     #0,     @#SBDDAT;COMPARE DATA
2262  012544  001402                          BEQ     24$             ;BRANCH IF GOOD
2263  012546  004737  012560                  JSR     PC,@#ERCS2C     ;JUMP TO ERROR FOR CLR (BIT 5)
2264                                                                  ;IN RHCS2
2265
2266
2267                                          ;LOOK-AHEAD REGISTER
2268
2269  012552  005720                  24$:    TST     (R0)+           ;AS THE LOOK-AHEAD REG. CANNOT BE PREDICTED
2270                                                                  ;AFTER AN INIT IT IS NOT CHECKED
2271
2272                                          ;CURRENT CYLINDER ADDRESS REGISTER
2273
2274  012554  005720                  25$:    TST     (R0)+           ;AS THE CURRENT REG. CANNOT BE PREDICTED
2275                                                                  ;AFTER A INIT IT IS NOT CHECKED
2276
2277  012556                          26$:
2278  012556  000405                          BR      TST25           ;BRANCH OVER JSR
2279
2280
2281  012560  014037  040172  ERCS2C: MOV     -(R0),  @#REGADR ;FAILING REGISTER ADDRESS
2282  012564  104001                  ERROR   1                       ;CLR (BIT 5) IN RHCS2 DID
2283                                                                  ;NOT CLEAR APPROPIATE BITS
2284                                                                  ;OR CLEARED EXTRA BITS
2285  012566  005720                          TST     (R0)+           ;UNDO -(R0) FOR BAD DATA
2286  012570  000207                          RTS     PC              ;RETURN TO TEST ABOVE
```

```
2287
2288
2289                                    ;*****************************************************************
2290                                    ;*TEST 25       PACK ACKNOWLEDGE COMMAND TEST
2291
2292                                    ;*      THE PACK ACKNOWLEDGE COMMAND WILL BE LOADED INTO RHCS1 WITH GO
2293                                    ;*      THEN ALL REGISTERS WILL BE CHECKED
2294                                    ;*      RH CLEAR WILL BE GIVEN
2295                                    ;*      THEN ALL REGISTERS WILL BE CHECKED
2296
2297                                    ;*****************************************************************
2298  012572  000004                    TST25:  SCOPE
2299  012574  012706  001000                    MOV     #STACK,SP       ;RESET STACK
2300  012600  012737  000025  002020            MOV     #TTNO,@#TSTNM   ;THIS SAVES TEST NUMBER
2301
2302  012606  004737  040522                    JSR     PC,@#CLDISK     ;INIT AND SET UP GENERAL CPU/DEVICE
2303                                                                    ;REGISTER CORRESPONDENCE AND UNIT NO.
2304  012612  012777  000001  167032            MOV     #DMD,@RHMR      ;SET DIAGNOSTIC MODE
2305  012620  013777  002062  167004            MOV     @#PKACK,@RHCS1  ;LOAD "PACK ACKNOWLEDGE COMMAND" INTO RHCS1
2306
2307                                            ;SAVE REGISTERS FOR COMPARISON AFTER 'GO' IS ISSUED
2308  012626  004037  041214                    JSR     R0,@#SAVER      ;SAVE
2309  012632  001624                            RHMC                    ;FROM
2310  012634  003134                            REINTO                  ;TO
2311  012636  000023                            19.                     ;NUMBER OF REGISTERS SAVED
2312
2313
2314  012640  052777  000001  166764            BIS     #GO,@RHCS1      ;ISSUE 'GO' TO PACK ACKNOWLEDGE COMMAND
2315
2316                                            ;CHANGE SAVED REGISTERS TO EXPECTED VALUES
2317  012646  052737  000100  003164            BIS     #VV,@#REINTO+30 ;SAVED RHDS1
2318
2319                                            ;AFTER GO HAS BEEN GIVEN TO PACK ACKNOWLEDGE COMMAND
2320                                            ;SAVE REGISTERS AGAIN SO THAT COMPARISONS CAN
2321                                            ;BE DONE
2322
2323  012654  004037  041214                    JSR     R0,@#SAVER      ;SAVE
2324  012660  001624                            RHMC                    ;FROM
2325  012662  002070                            WRFROM
2326  012664  000023                            19.                     ;NUMBER OF REGISTERS SAVED
2327
2328                                            ;AS UPPER BYTE OF RHAS CAN BE CHANGING IN A DUAL PORT
2329                                            ;OPERATION THE UPPER BYTE OF RHAS WILL BE SAVED AS IS
2330                                            ;SO THAT THE COMPARES ARE ONLY VALID FOR THE LOWER BYTE
2331  012666  113737  003161  002115            MOVB    @#REINTO+25,@#WRFROM+25;SAVE UPPER RHAS
2332
2333
2334                                            ;COMPARE REGISTERS BEFORE PACK ACKNOWLEDGE COMMAND
2335                                            ;WITH AFTER GO
2336
2337  012674  004037  041416                    JSR     R0,@#COMPAR     ;COMPARE
2338  012700  003134                            REINTO                  ;GOOD BUFFER
2339  012702  002070                            WRFROM                  ;TEST BUFFER
2340  012704  000023                            19.                     ;NUMBER
```

```
2341  012706  012714                        1$                      ;RETURN FOR ERROR
2342  012710  012714                        1$                      ;SAME
2343  012712  012734                        2$                      ;RETURN FOR GOOD COMPARISON
2344
2345  012714  013705  045272          1$:   MOV     @#ERWORD,R5       ;GETTING READY TO INDEX
2346  012720  060505                        ADD     R5,R5             ;DOUBLE ERROR WORD
2347  012722  016537  001622  040172        MOV     RHWC-2(R5),@#REGADR ;FAILING REGISTER ADDRESS
2348
2349  012730  104001                        ERROR   1                 ;IMPROPER REGISTER CHANGE
2350                                                                  ;AFTER PACK ACKNOWLEDGE COMMAND
2351                                                                  ;WITH GO IS GIVEN
2352  012732  000207                        RTS     PC                ;RETURN TO COMPARISON
2353
2354  012734                          2$:                             ;CONTINUE WITH THE NEXT TEST
```

```
                                        ;*****************************************************************
                                        ;*TEST 26        UNIBUS INIT TEST
                                        ;*
                                        ;*      ALL POSSIBLE REGISTER BITS ARE FILLED WITH ONES
                                        ;*      A RESET COMMAND IS GIVEN
                                        ;*      ALL REGISTERS ARE CHECKED
                                        ;*
                                        ;*****************************************************************
012734  000004                    TST26:  SCOPE
012736  012706  001000                    MOV     #STACK,SP       ;RESET STACK
012742  012737  000026  002020            MOV     #TTNO,@#TSTNM   ;THIS SAVES TEST NUMBER
012750  004737  040522                    JSR     PC,@#CLDISK     ;INIT AND SET UP GENERAL CPU/DEVICE
                                                                ;REGISTER CORRESPONDENCE

                                        ;FILL ALL POSSIBLE REGISTER BITS WITH ONES

012754  012777  177777  166642          MOV     #177777,@RHWC   ;WORD COUNT REGISTER GETS 177777
012762  012777  177777  166636          MOV     #177777,@RHBA   ;BUS ADDRESS REGISTER GETS 177777
012770  052777  157010  166632          BIS     #157010,@RHCS2  ;CONTROL AND STATUS 2 GETS 177430
012776  012777  001476  166626          MOV     #1476,@RHCS1    ;CONTROL AND STATUS REGISTER 1 GETS 21476
013004  012777  177777  166622          MOV     #177777,@RHER1  ;ERROR REGISTER1 GETS 177777
013012  012777  017437  166616          MOV     #17437,@RHDST   ;DESIRED SECTOR TRACK
013020  012777  177777  166612          MOV     #177777,@RHER2  ;ERROR REGISTER 2
013026  012777  016277  166606          MOV     #16277,@RHOF    ;OFFSET REGISTER
013034  012777  000777  166602          MOV     #777,@RHCA      ;DESIRED CYLINDER
013042  012777  177777  166576          MOV     #177777,@RHER3  ;ERROR REGISTER 3
013050  012777  000001  166574          MOV     #DMD,@RHMR      ;MAINTENANCE REGISTER
013056  012777  177777  166566          MOV     #177777,@RHMR   ;MAINTENANCE REGISTER

                                        ;BEFORE RESET SAVE REGISTERS IN READ INTO BUFFER
013064  004037  041214                  JSR     R0,@#SAVER      ;SAVE
013070  001624                          RHWC                    ;FROM
013072  003134                          REINTO                  ;TO
013074  000021                          17.                     ;NUMBER

                                        ;GIVE RESET AND REINSTATE UNIT NUMBER
013076  000005                          RESET
013100  004737  052512                  JSR     PC,@#STKINT     ;INITIALIZE TK
013104  053777  001762  166516          BIS     @#UNIT,@RHCS2

                                        ;CHANGE ORIGINAL SAVED REGISTERS TO EXPECTED VALUES AFTER RESET
013112  005037  003136                  CLR     @#REINTO+2      ;CLEAR SAVED RHBA
013116  013746  001762                  MOV     @#UNIT,-(SP)    ;GET UNIT NUMBER FRO SAVED RHCS2
013122  052716  000100                  BIS     #IR,(SP)        ;INCLUDE IR
013126  012637  003140                  MOV     (SP)+,@#REINTO+4 ;SAVED RHCS2
013132  012737  004276  003142          MOV     #DVA!RDY!76,@#REINTO+6 ;SAVED RHCS1
013140  005037  003144                  CLR     @#REINTO+10     ;SAVED RHER1
013144  005037  003150                  CLR     @#REINTO+14     ;SAVED RHER2
013150  012737  116000  003152          MOV     #116000,@#REINTO+16 ;SAVED RHOF
013156  005037  003156                  CLR     @#REINTO+22     ;SAVED RHER3
013162  105037  003160                  CLRB    @#REINTO+24     ;SAVED RHAS
```

```
2409  013166  012737  000400  003162          MOV     #400,@#REINTO+26;SAVED RHMR
2410
2411                                          ;CHANGE RHDS1 WITHOUT CHANGING PROG BIT
2412  013174  013746  003164                  MOV     @#REINTO+30,-(SP) ;GET RHDS1
2413  013200  042716  176777                  BIC     #^CPROG,(SP)    ;CLEAR EVERYTHING EXCEPT PROG
2414  013204  052716  000700                  BIS     #700,(SP)       ;SET EXPECTED BITS - 'DPR', 'DRY' & 'VV'
2415
2416  013210  012637  003164          4$:     MOV     (SP)+,@#REINTO+30;SAVED RHDS1
2417  013214  005037  003172                  CLR     @#REINTO+36     ;SAVED RHEC1
2418  013220  005037  003174                  CLR     @#REINTO+40     ;SAVED RHEC2
2419
2420                                          ;AFTER RESET, SAVE REGISTERS FOR COMPARISONS TO BE DONE
2421  013224  004037  041214                  JSR     R0,@#SAVER      ;SAVE
2422  013230  001E24                          RHMC                    ;FROM
2423  013232  002070                          WRFROM                  ;TO
2424  013234  000021                          17.                     ;NUMBER
2425
2426                                          ;AS UPPER BYTE OF RHAS CAN BE CHANGING IN A DUAL PORT
2427                                          ;OPERATION THE UPPER BYTE OF RHAS WILL BE SAVED AS IS
2428                                          ;SO THAT THE COMPARES ARE ONLY VALID FOR THE LOWER BYTE
2429  013236  113737  003161  002115          MOVB    @#REINTO+25,@#WRFROM+25;SAVE UPPER RHAS
2430
2431                                          ;COMPARE REGISTERS BEFORE RESET WITH REGISTERS AFTER RESET
2432  013244  004037  041416                  JSR     R0,@#COMPAR     ;COMPARE
2433  013250  003134                          REINTO                  ;GOOD BUFFER
2434  013252  002070                          WRFROM                  ;TEST BUFFER
2435  013254  000021                          17.                     ;NUMBER
2436  013256  013264                          1$                      ;RETURN FOR ERROR
2437  013260  013264                          1$                      ;SAME
2438  013262  013304                          2$                      ;RETURN FOR GOOD COMPARISON
2439
2440  013264  013705  045272          1$:     MOV     @#ERWORD,R5     ;GETTING READY TO INDEX
2441  013270  060505                          ADD     R5,R5           ;DOUBLE ERROR WORD
2442  013272  016537  001622  040172          MOV     RHMC-2(R5),@#REGADR ;FAILING REGISTER ADDRESS
2443  013300  104001                          ERROR   1               ;REGISTER CONTENTS AFTER
2444                                                                  ;A RESET THAT IS AN
2445                                                                  ;UNIBUS INITIALIZE CAUSED
2446                                                                  ;AN IMPROPER REGISTER CHANGE
2447  013302  000207                          RTS     PC              ;RETURN TO COMPARISON
2448  013304                          2$:                             ;RETURN TO POINT ON GOOD COMPARISON
```

```
                                        .SBTTL  SILO TESTS


                                ;*****************************************************************
                                ;*TEST 27        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

                                ;*****************************************************************
013304  000004                  TST27:  SCOPE
013306  012706  001000                  MOV     #STACK,SP       ;RESET STACK
013312  012737  000027  002020          MOV     #TTNO,@#TSTNM   ;THIS SAVES TEST NUMBER
013320  004767  025176                  JSR     PC,CLDISK       ;CLEAR DISK AND LOAD R'S
013324  017700  166272                  MOV     @RHDB,R0        ;READ FROM EMPTY SILO
013330  013746  001762                  MOV     @#UNIT,-(SP)    ;GET UNIT NO. IN
013334  052716  100100                  BIS     #DLT!IR,(SP)    ;GET DATA LATE BIT AND IR
013340  004737  040126                  JSR     PC,@#PUTREG     ;SAVE REGISTERS
013344  022637  001700                  CMP     (SP)+,@#CS2     ;IS DATA LATE BIT UP?
013350  001403                          BEQ     1$              ;IF YES BRANCH
013352  010237  001122                  MOV     R2,@#SBDADR     ;IF NOT STORE FAILING REG.
013356  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
013360  022737  144200  001702  1$:     CMP     #SC!TRE!RDY!DVA,@#CS1 ;IS SPECIAL CONDITION, TRANSFER ERROR
                                                                ;AND READY UP?
013366  001403                          BEQ     2$              ;IF YES BRANCH
013370  010137  001122                  MOV     R1,@#SBDADR     ;IF NOT STORE FAILING REG.
013374  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
013376  012711  040000          2$:     MOV     #TRE,@R1        ;CLEAR ERROR BITS BY MOVING
                                                                ;ONE INTO TRE IN RHCS1
013402  004737  040126                  JSR     PC,@#PUTREG     ;SAVE REGISTERS
013406  022737  004200  001702          CMP     #RDY!DVA,@#CS1  ;ALL BITS BUT RDY AND DVA SHOULD
                                                                ;BE 0
013414  001403                          BEQ     3$              ;BRANCH IF YES
013416  010137  001122                  MOV     R1,@#SBDADR     ;STORE FAILING ADDRESS
013422  104011                          ERROR   11              ;AFTER A ONE IN TRE ONLY
                                                                ;READY AND DVA SHOULD BE SET IN
                                                                ;RHCS1
013424  013746  001762          3$:     MOV     @#UNIT,-(SP)
013430  052716  000100                  BIS     #IR,(SP)
013434  022637  001700                  CMP     (SP)+,@#CS2     ;RHCS2 SHOULD HAVE IR AND UNIT ONLY
013440  001403                          BEQ     TST30           ;BRANCH IF YES
013442  010237  001122                  MOV     R2,@#SBDADR     ;STORE FAILING ADDR
```

```
 2503  013446  104011                          ERROR   11              ;AFTER A ONE IN TRE ONLY
```

```
2504
2505
2506                                    ;*****************************************************************
2507                                    ;*TEST 30       SILO TEST 2
2508
2509                                    ;*      THIS TESTS THE IR AND "OR" BITS OF RHCS2
2510                                    ;*      AT THE BEGINNING IR SHOULD BE SET AND "OR" RESET
2511                                    ;*      LOADING 0 IN SILO RESETS IR FOR ONLY 2 MICRO SECONDS
2512                                    ;*      THIS TIME CANNOT BE CHECKED BUT IT IS CHECKED TO SEE IF
2513                                    ;*      IT DOES GO DOWN OR NOT
2514                                    ;*      THEN ALL 1 IS LOADED IN SILO "OR" SHOULD BECOME SET
2515                                    ;*      IN 30 MICRO SECONDS AGAIN TIME IS NOT  CHECKED
2516                                    ;*      "OR" SHOULD BE SET
2517                                    ;*      THE OUTPUT FROM THE SILO SHOULD BE 0 AND ALL ONES
2518
2519                                    ;*****************************************************************
2520  013450  000004                    TST30:  SCOPE
2521  013452  012706  001000                    MOV     #STACK,SP       ;RESET STACK
2522  013456  012737  000030  002020            MOV     #TTNO,@#TSTNM   ;THIS SAVES TEST NUMBER
2523  013464  004767  025032                    JSR     PC,CLDISK       ;CLEAR REGISTERS LOAD R'S
2524  013470  013746  001762                    MOV     @#UNIT,-(SP)
2525  013474  052716  000100                    BIS     #IR,(SP)
2526  013500  004737  040126                    JSR     PC,@#PUTREG     ;SAVE REGISTERS
2527  013504  022637  001700                    CMP     (SP)+,@#CS2     ;IR SHOULD BE SET "OR" RESET
2528  013510  001403                            BEQ     1$
2529  013512  010237  001122                    MOV     R2,@#SBDADR     ;FAILING REGISTER RHCS2
2530  013516  104011                            ERROR   11              ;RHCS2 DOES NOT HAVE IR
2531                                                                    ;SET, UNIT NO. SET AND
2532                                                                    ;ALL OTHER BITS 0
2533  013520  005077  166076            1$:     CLR     @RHDB           ;LOAD DATA BUFFER (SILO) WITH 0
2534  013524  012777  177777  166070            MOV     #-1,@RHDB       ;LOAD SILO WITH ALL ONES
2535  013532  013737  001630  013542            MOV     @#RHCS2,@#2$    ;ADDRESS OF RHCS2
2536  013540  104424                            WAT                     ;WAIT TRAP
2537  013542  000000            2$:             .WORD                   ;ADDRESS OF RHCS2
2538  013544  000200                            OR
2539  013546  013746  001762            3$:     MOV     @#UNIT,-(SP)    ;
2540  013552  052716  000300                    BIS     #OR!IR,(SP)     ;IR AND "OR"
2541  013556  004737  040126                    JSR     PC,@#PUTREG     ;SAVE REGISTERS
2542  013562  022637  001700                    CMP     (SP)+,@#CS2     ;IR AND "OR" SHOULD BE SET
2543  013566  001403                            BEQ     4$
2544  013570  010237  001122                    MOV     R2,@#SBDADR     ;SAVE RHCS2 ADDR. FAILING REG.
2545  013574  104011                            ERROR   11              ;"OR" IN RHCS2 SHOULD BE
2546                                                                    ;SET TOGETHER WITH I.R AND
2547                                                                    ;UNIT NO.
2548  013576  017700  166020            4$:     MOV     @RHDB,R0        ;SAVE SILO DATA SHOULD BE 0
2549  013602  017705  166014                    MOV     @RHDB,R5        ;SAVE SILO DATA SHOULD BE ALL 1
2550  013606  022700  000000                    CMP     #0,R0           ;FIRST WORD 0? XYZ DO MORE TEST
2551  013612  001410                            BEQ     5$              ;BRANCH IF YES
2552  013614  005037  001124                    CLR     @#SGDDAT        ;GOOD DATA
2553
2554  013620  010037  001126                    MOV     R0,@#SBDDAT     ;BAD DATA
2555  013624  013737  001622  040172            MOV     @#RHDB,@#REGADR ;SAVE RHDB FAILING REG.
2556  013632  104001                            ERROR   1               ;SILO DID NOT HAVE THE FIRST WORD
2557                                                                    ;"0" WHEN "OR" WAS SET
```

M08


```
2558  013634  022705  177777          55:     CMP     #-1,R5          ;SECOND WORD ALL ONES?
2559  013640  001411                          BEQ     TST31           ;BRANCH IF YES
2560  013642  012737  177777  001124          MOV     #-1,@#$GDDAT    ;GOOD DATA
2561  013650  010537  001126                  MOV     R5,@#$BDDAT     ;BAD DATA
2562  013654  013737  001622  040172          MOV     @#RHDB,@#REGADR ;SAVE RHDB FAILING REG.
2563  013662  104001                          ERROR   1               ;SILO DID NOT HAVE THE SECOND
2564                                                                  ;WORD OF ALL ONES WHEN "OR"
2565                                                                  ;WAS SET
2566
```

```
2567
2568
2569                                    ;*****************************************************************
2570                                    ;*TEST 31        SILO TEST 3
2571
2572                                    ;*      THIS TESTS SILO BUFFER BY FILLING IT WITH A COUNT FROM
2573                                    ;*      0 TO 65 AND THEN CHECKING IF IR IS DOWN AND "OR"
2574                                    ;*      IS HIGH AND COMPARING THE SILO OUTPUT.
2575
2576                                    ;*****************************************************************
2577  013664  000004                   TST31:  SCOPE
2578  013666  012737  000031  002020           MOV     #TTNO,@#TSTNM   ;THIS SAVES TEST NUMBER
2579  013674  012700  047070                   MOV     #SILOTB,R0      ;TABLE POINTER
2580  013700  012705  000103                   MOV     #67.,R5         ;COUNTER
2581  013704  005020                   1$:     CLR     (R0)+           ;CLEAR TOTAL TABLE
2582  013706  005305                           DEC     R5              ;COUNT
2583  013710  001375                           BNE     1$              ;BRANCH IF NOT COMPLETELY CLEAR
2584  013712  004767  024604                   JSR     PC,CLDISK       ;CLEAR ALL REG.
2585  013716  005000                           CLR     R0
2586  013720  012705  000102                   MOV     #66.,R5         ;COUNT
2587  013724  010077  165672           2$:     MOV     R0,@RHDB        ;LOAD SILO WITH COUNT FROM 0 TO 65
2588  013730  005200                           INC     R0              ;NEXT COUNT
2589  013732  005305                           DEC     R5              ;IS 66 LOADS DONE?
2590  013734  001373                           BNE     2$              ;BRANCH IF NOT.
2591  013736  013746  001762                   MOV     @#UNIT,-(SP)
2592  013742  052716  000200                   BIS     #OR,(SP)
2593  013746  004737  040126                   JSR     PC,@#PUTREG     ;SAVE REGISTERS
2594  013752  022637  001700                   CMP     (SP)+,@#CS2     ;"OR" SHOULD BE SET IR RESET
2595  013756  001405                           BEQ     3$              ;BRANCH IF YES
2596  013760  010237  001122                   MOV     R2,@#$BDADR     ;SAVE RHCS2 ADR. FAILING REG.
2597  013764  104011                           ERROR   11              ;"OR" WAS NOT SET, IR WAS NOT
2598  013766  005037  001774                   CLR     @#ERFLG$        ;RESET AFTER SILO WAS FULL
2599  013772  012700  047070           3$:     MOV     #SILOTB,R0      ;POINTER
2600  013776  012705  000102                   MOV     #66.,R5         ;COUNTER
2601  014002  017720  165614           4$:     MOV     @RHDB,(R0)+     ;READ SILO
2602  014006  005305                           DEC     R5              ;COUNT
2603  014010  001374                           BNE     4$              ;BRANCH IF 66 NOT DONE
2604  014012  012700  047070                   MOV     #SILOTB,R0      ;POINTER
2605  014016  012705  000102                   MOV     #66.,R5
2606  014022  005046                           CLR     -(SP)
2607  014024  021620                   5$:     CMP     (SP),(R0)+
2608  014026  001425                           BEQ     7$              ;BRANCH IF GOOD
2609  014030  014037  001126                   MOV     -(R0),@#$BDDAT  ;BAD DATA
2610  014034  011637  001124                   MOV     (SP),@#$GDDAT   ;GOOD DATA
2611  014040  013737  001622  040172           MOV     @#RHDB,@#REGADR ;FAILING REG. RHDB
2612  014046  005737  001774                   TST     @#ERFLG$        ;IS THIS FIRST ERROR?
2613  014052  001002                           BNE     6$              ;IF NOT BRANCH
2614  014054  104012                           ERROR   12              ;THESE TWO ERROR CALLS ARE FOR
2615  014056  000401                           BR      64$             ;BRANCH TO AVOID PRINTING NEXT ERROR
2616  014060  104013                   6$:     ERROR   13              ;THE SAME TYPEOUT.  SILO
2617                                           ;HAD A COUNT WRITTEN IN.
2618                                           ;ON READ OUT AN ERROR WAS
2619                                           ;DETECTED. THE TOTAL SILO
2620                                           ;READOUT IS IN LOCATION
```

```
2621                                                    ;"SILOTB" TO THE NEXT 65
2622                                                    ;WORDS.
2623  014062  005720              64$:    TST     (R0)+ ;INCREMENT (R0)
2624                                                    ;ARE FURTHER COMPARES TO
2625  014064  017746  163500              MOV     DSKS,-(SP) ;BE DONE
2626  014070  042716  177577              BIC     #!CSW07!SW08,(SP) ;ONLY KEEP SW7 AND SW8
2627  014074  022726  000200              CMP     #SW07,(SP)+ ;TEST SW07
2628  014100  001403                      BEQ     10$     ;IF NO MORE COMPARE THEN BRANCH
2629  014102  005216              7$:     INC     (SP)    ;NEXT GOOD WORD
2630  014104  005305                      DEC     R5      ;COUNT
2631  014106  001346                      BNE     5$      ;BRANCH IF 66 NOT COMPLETE
2632  014110  005726              10$:    TST     (SP)+   ;POP STACK
```

```
2633
2634
2635
2636                                    ;*****************************************************************
2637                                    ;*TEST 32       SILO TEST4
2638                                    ;*      NOW PUT 67 WORDS INTO SILO AND CHECK FOR DLT
2639                                    ;*      EVEN AFTER THE 67TH. WORD INPUT THE FIRST WORD SHOULD NOT CHANGE
2640
2641                                    ;*****************************************************************
2642  014112  000004                    TST32:  SCOPE
2643  014114  012737  000032  002020            MOV     #TTNO,@#TSTNM   ;THIS SAVES TEST NUMBER
2644  014122  004767  024374                    JSR     PC,CLDISK       ;CLEAR DISK REG.
2645  014126  005000                            CLR     R0              ;CLEAR R0
2646  014130  005200                    1$:     INC     R0              ;ADD 1
2647  014132  010077  165464                    MOV     R0,@RHDB        ;LOAD SILO
2648  014136  022700  000103                    CMP     #67.,R0         ;67 DONE?
2649  014142  001401                            BEQ     2$              ;BRANCH IF YES
2650  014144  000771                            BR      1$              ;NO SO BRANCH
2651  014146  004737  040126            2$:     JSR     PC,@#PUTREG     ;SAVE REGISTERS
2652
2653  014152  032737  100000  001700            BIT     #DLT,@#CS2      ;DLT SET?
2654  014160  001003                            BNE     3$              ;BRANCH IF YES
2655  014162  010237  001122                    MOV     R2,@#$BDADR     ;FAILING ADDRESS RHCS2
2656  014166  104011                            ERROR   11              ;DATA LATE DID NOT SET AT 67TH.
2657  014170  017737  165426  001126    3$:     MOV     @RHDB,@#$BDDAT  ;INPUT TO SILO
2658  014176  012737  000001  001124            MOV     #1,@#$GDDAT     ;GOOD DATA
2659  014204  023737  001124  001126            CMP     @#$GDDAT,@#$BDDAT ;COMPARE
2660  014212  001404                            BEQ     TST33           ;BRANCH IF GOOD
2661  014214  013737  001622  040172            MOV     @#RHDB,@#REGADR ;FAILING REG. RHDB
2662  014222  104012                            ERROR   12              ;WORD IN RHDB CHANGED
2663                                            ;AFTER THE 67TH INPUT.
```

```
                                ;*****************************************************************
                                ;*TEST 33       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 ARE DONE, ON THE LAST, 'DTL' IN RHCS2 SHOULD BE SET.
                                ;
                                ;*****************************************************************
014224  000004                  TST33:  SCOPE
014226  012737  000033  002020          MOV     #TTNO,@#TSTNM   ;THIS SAVES TEST NUMBER
014234  004737  040522                  JSR     PC,@#CLDISK     ;CLEAR DISK
014240  013746  001762                  MOV     @#UNIT,-(SP)    ;GET UNIT NO.
014244  052716  000100                  BIS     #IR,(SP)        ;SET INPUT READY
014250  004737  040126                  JSR     PC,@#PUTREG     ;SAVE REGISTERS
014254  022637  001700                  CMP     (SP)+,@#CS2     ;IR SHOULD BE SET "OR" CLEARED
014260  001403                          BEQ     1$              ;BRANCH IF GOOD
014262  010237  001122                  MOV     R2,@#SBDADR     ;FAILING REGISTER RHCS2
014266  104011                          ERROR   11              ;RHCS2 DOES NOT HAVE IR SET
                                                                ;AND ALL OTHER BITS 0
014270  013700  001622          1$:     MOV     @#RHDB,R0       ;R0 HAS RHDB ADDRESS
014274  005001                          CLR     R1              ;DATA
014276  010110                  2$:     MOV     R1,@R0          ;0, THEN 1 THEN 2 THEN 3
                                                                ;IN RHDB
014300  005201                          INC     R1              ;INCREMENT DATA
014302  022701  000004                  CMP     #4,R1           ;IS 4 DONE
014306  103373                          BHIS    2$              ;BRANCH IF NOT
014310  013737  001630  014320          MOV     @#RHCS2,@#3$
014316  104424                          WAT                     ;WAIT FOR "OR"
014320  000000                  3$:     .WORD   0               ;RHCS2 ADDRESS
014322  000200                          OR                      ;WAIT ON OR,
014324  004737  040522                  JSR     PC,@#CLDISK     ;CLR IN RHCS2
014330  013746  001762                  MOV     @#UNIT,-(SP)    ;UNIT NO.
014334  052716  000100                  BIS     #IR,(SP)
014340  004737  040126                  JSR     PC,@#PUTREG     ;SAVE REGISTERS
014344  022637  001700                  CMP     (SP)+,@#CS2     ;IR SHOULD BE SET "0"=0
014350  001403                          BEQ     4$              ;BRANCH IF GOOD
014352  010237  001122                  MOV     R2,@#SBDADR     ;FAILING REGISTER RHCS2
014356  104011                          ERROR   11              ;RHCS2 DOES NOT HAVE IR SET
                                                                ;AND ALL OTHER BITS 0
014360  013700  001622          4$:     MOV     @#RHDB,R0       ;R0 HAS RHDB ADDRESS
014364  012710  000004                  MOV     #4,@R0          ;LOAD 4 IN SILO
014370  011201                          MOV     @R2,R1          ;SAVE RHCS2
014372  011005                          MOV     @R0,R5          ;READ THE 4 IN SILO
014374  011003                          MOV     @R0,R3          ;READ SILO TO GET DLT
014376  011204                          MOV     @R2,R4          ;SAVE RHCS2
014400  032701  000200                  BIT     #OR,R1          ;TEST FOR OR IN RHCS2
014404  001424                          BEQ     6$              ;IF OR IS NOT SET BRANCH
014406  022705  000004                  CMP     #4,R5           ;SILO 4 IS NOW COMPARED
014412  001410                          BEQ     5$
014414  010037  040172                  MOV     R0,@#REGADR     ;SILO ADDRESS
```

E09


```
2718  014420  012737  000004  001124          MOV     #4,@#$GDDAT     ;GOOD DATA
2719  014426  010537  001126                  MOV     R5,@#$BDDAT     ;BAD DATA
2720  014432  104001                          ERROR   1               ;SILO DID NOT CONTAIN WORD
2721                                                                  ;PUT IN AFTER "OR" WAS UP
2722  014434  005703                  5$:     TST     R3              ;IS IT ZERO BECAUSE SILO
2723                                                                  ;IS DESTRUCTIVE READ
2724  014436  001407                          BEQ     6$              ;BRANCH IF GOOD
2725  014440  010037  040172                  MOV     R0,@#REGADR     ;SILO ADDRESS
2726  014444  005037  001124                  CLR     @#$GDDAT        ;GOOD DATA
2727  014450  010337  001126                  MOV     R3,@#$BDDAT     ;BAD DATA
2728  014454  104001                          ERROR   1               ;SILO SHOULD BE ZERO
2729                                                                  ;AFTER THE ONE WORD PUT IN
2730                                                                  ;HAS BEEN TAKEN OUT AS
2731                                                                  ;SILO IS A DESTRUCTIVE READ
2732  014456  032704  100000          6$:     BIT     #DLT,R4         ;
2733  014462  001013                          BNE     TST34           ;BRANCH IF DLT SET
2734  014464  013746  001762                  MOV     @#UNIT,-(SP)    ;GET UNIT NO
2735  014470  052716  100300                  BIS     #DLT!DR!IR,(SP) ;
2736  014474  012637  001124                  MOV     (SP)+,@#$GDDAT  ;GOOD DATA
2737  014500  010437  001126                  MOV     R4,@#$BDDAT     ;BAD DATA
2738  014504  010237  040172                  MOV     R2,@#REGADR     ;RHCS2 ADDRESS
2739  014510  104001                          ERROR   1               ;DATA LATE ERROR
```

```
                                .SBTTL  MORE REGISTER TESTS

                                ;*****************************************************************
                                ;*TEST 34        TEST ODD BYTE INSTRUCTION ON RHCS1
                                ;*
                                ;*      RDY (BIT 07) AND DVA (BIT 11) SHOULD ALWAYS BE SET
                                ;*
                                ;*****************************************************************
014512  000004                  TST34:  SCOPE
014514  012737  000034  002020          MOV     #TTNO,@#TSTNM   ;THIS SAVES TEST NUMBER
014522  012706  001000          MOV     #STACK,SP       ;RESET STACK
014526  004767  023770          JSR     PC,CLDISK       ;CLEAR DISK REG.
014532  012711  003566          MOV     #3566,@R1       ;LOAD RHCS1 WITH ANY NUMBER
014536  010146                  MOV     R1,-(SP)        ;GETTING READY TO FORM ODD BYTE
014540  005216                  INC     (SP)            ;SP NOW HAS ODD BYTE FOR RHCS1
014542  112736  000005          MOVB    #5,@(SP)+       ;MOVE 5 INTO ODD BYTE FOR RHCS1
014546  011137  001126          MOV     @R1,@#$BDDAT    ;TEST DATA
014552  022737  006766  001126  CMP     #2566!DVA!RDY,@#$BDDAT ;RHCS1 SHOULD HAVE 6766
014560  001406                  BEQ     1$              ;BRANCH IF GOOD
014562  012737  006766  001124  MOV     #2566!DVA!RDY,@#$GDDAT ;GOOD DATA
014570  010137  040172          MOV     R1,@#REGADR     ;FAILING REGISTER RHCS1
014574  104001                  ERROR   1               ;MOVING A NUMBER INTO
                                                        ;ODD BYTE OF RHCS1 GAVE
                                                        ;WRONG RESULTS
014576  112711  000032  1$:     MOVB    #32,@R1         ;MOVE INTO EVEN BYTE
014602  011137  001126          MOV     @R1,@#$BDDAT    ;TEST DATA
014606  022737  006632  001126  CMP     #2432!DVA!RDY,@#$BDDAT ;RHCS1 SHOULD HAVE 6632
014614  001406                  BEQ     TST35           ;;BRANCH IF GOOD
014616  012737  006632  001124  MOV     #2432!DVA!RDY,@#$GDDAT ;GOOD DATA
014624  010137  040172          MOV     R1,@#REGADR     ;FAILING REGISTER RHCS1
014630  104001                  ERROR   1               ;MOVING A NUMBER INTO EVEN
                                                        ;BYTE OF RHCS1 GAVE WRONG
                                                        ;RESULTS
```

```
2777
2778
2779
2780                                    ;*****************************************************************
2781                                    ;*TEST 35       TEST ODD BYTE INSTRUCTION ON RHCS2
2782
2783                                    ;*      IR (BIT 06) AND THE UNIT SELECT (BIT 0-2) WILL BE SET
2784
2785                                    ;*****************************************************************
2786  014632  000004                    TST35:  SCOPE
2787
2788  014634  012737  000035  002020            MOV     #TTNO,@#TSTNM   ;THIS SAVES TEST NUMBER
2789  014642  004767  023654                    JSR     PC,CLDISK
2790  014646  052712  177000                    BIS     #177000,(R2)    ;LOAD RHCS2
2791  014652  010246                            MOV     R2,-(SP)        ;GETTING READY FOR ODD BYTE
2792  014654  005216                            INC     (SP)            ;SP NOW HAS ODD BYTE FOR RHCS2
2793  014656  105036                            CLRB    @(SP)+          ;CLERR RHCS2 ODD BYTE
2794  014660  013746  001762                    MOV     @#UNIT, -(SP)   ;GET UNIT NO.
2795  014664  052716  000100                    BIS     #IR,(SP)        ;INPUT READY AS IT IS SET
2796  014670  011237  001126                    MOV     @R2,@#$BDDAT    ;TEST DATA
2797  014674  022637  001126                    CMP     (SP)+,@#$BDDAT  ;COMPARE TO SEE THAT
2798                                                                    ;"CLRB" DID CLEAR
2799  014700  001411                            BEQ     1$
2800  014702  013737  001762  001124            MOV     @#UNIT, @#$GDDAT
2801  014710  052737  000100  001124            BIS     #IR,    @#$GDDAT ;GOOD DATA
2802  014716  010237  040172                    MOV     R2,     @#REGADR ;FAILING REGISTER RHCS2
2803  014722  104001                            ERROR   1               ;CLEARING ODD BYTE OF RHCS2
2804                                                                    ;GAVE WRONG RESULTS
2805  014724  013746  001762          1$:       MOV     @#UNIT, -(SP)
2806  014730  052716  000010                    BIS     #BAI,   (SP)
2807  014734  052712  020000                    BIS     #UPE,@R2        ;HAVE UPE AND MPE IN RHCS2
2808                                                                    ;BESIDES UNIT SELECT
2809  014740  112612                            MOVB    (SP)+,  @R2     ;MOVE INTO EVEN BYTE OF RHCS2
2810  014742  013746  001762                    MOV     @#UNIT, -(SP)
2811  014746  052716  020110                    BIS     #UPE!IR!BAI,(SP)
2812  014752  011637  001124                    MOV     (SP),   @#$GDDAT ;GOOD DATA
2813  014756  011237  001126                    MOV     @R2,@#$BDDAT    ;TEST DATA
2814  014762  022637  001126                    CMP     (SP)+,@#$BDDAT  ;COMPARE TO SEE THAT MOVB DID
2815                                                                    ;MOVE EVEN BYTE ONLY
2816  014766  001403                            BEQ     TST36           ;;BRANCH IF GOOD
2817  014770  010237  040172                    MOV     R2,@#REGADR     ;FAILING REGISTER RHCS2
2818  014774  104001                            ERROR   1               ;MOVING A NUMBER INTO EVEN
2819                                                                    ;BYTE OF RHCS2 GAVE WRONG
2820                                                                    ;RESULTS
2821
```

```
2822
2823
2824                                    ;*****************************************************************
2825                                    ;*TEST 36        ODD BYTE TEST ON RHWC
2826
2827                                    ;*      IN THIS REGISTER NO BITS SHOULD BE PERMANENTLY SET
2828
2829                                    ;*****************************************************************
2830  014776  000004                    TST36:  SCOPE
2831
2832  015000  012737  000036  002020            MOV     #TTNO,@#TSTNM   ;THIS SAVES TEST NUMBER
2833  015006  012706  001000                    MOV     #STACK,SP       ;RESET STACK
2834  015012  004767  023504                    JSR     PC,CLDISK       ;CLEAR DISK REGISTERS
2835  015016  013704  001624                    MOV     @#RHWC, R4      ;R4 NOW IS WORD COUNT REGISTER
2836  015022  012714  025252                    MOV     #25252, @R4     ;LOAD RHWC
2837  015026  010446                            MOV     R4,     -(SP)   ;GETTING READY TO FORM ODD BYTE
2838  015030  005216                            INC     (SP)            ;SP NOW HAS ODD BYTE FOR RHWC
2839  015032  112736  000377                    MOVB    #377,   @(SP)+  ;MOVE 377 INTO ODD BYTE OF RHWC
2840  015036  011437  001126                    MOV     @R4,@#$BDDAT    ;TEST DATA
2841  015042  022737  177652  001126            CMP     #177652,@#$BDDAT;COMPARE TO SEE IF MOVB DID OK
2842  015050  001406                            BEQ     1$              ;BRANCH IF GOOD
2843  015052  012737  177652  001124            MOV     #177652,@#$GDDAT ;GOOD DATA
2844  015060  010437  040172                    MOV     R4,     @#REGADR ;REGISTER FAILING RHWC
2845  015064  104001                            ERROR   1               ;MOVING INTO ODD BYTE OF RHWC
2846                                                                    ;GAVE WRONG RESULTS
2847  015066  112714  000123            1$:     MOVB    #123,   @R4     ;MOVE INTO EVEN BYTE OF RHWC
2848  015072  011437  001126                    MOV     @R4,@#$BDDAT    ;TEST DATA
2849  015076  022737  177523  001126            CMP     #177523,@#$BDDAT
2850  015104  001406                            BEQ     TST37           ;;BRANCH IF GOOD
2851  015106  012737  177523  001124            MOV     #177523,@#$GDDAT ;GOOD DATA
2852  015114  010437  040172                    MOV     R4,     @#REGADR ;REGISTER FAILING RHWC
2853  015120  104001                            ERROR   1
```

```
2854
2855
2856
2857                                    ;*****************************************************************
2858                                    ;*TEST 37       TEST ODD BYTE INSTRUCTION ON RHBA
2859
2860                                    ;*      BIT 0 SHOULD ALWAYS BE 0
2861
2862                                    ;*****************************************************************
2863  015122  000004                    TST37:  SCOPE
2864
2865  015124  012706  001000                    MOV     #STACK,SP       ;RESET STACK
2866  015130  012737  000037  002020            MOV     #TTNO,@#TSTNM   ;THIS SAVES TEST NUMBER
2867  015136  004767  023360                    JSR     PC,CLDISK
2868  015142  013704  001626                    MOV     @#RHBA, R4      ;R4 HAS ADDRESS OF RHBA
2869  015146  012714  025253                    MOV     #25253, @R4     ;LOAD RHBA
2870  015152  010446                            MOV     R4,     -(SP)   ;GETTING READY FOR ODD BYTE
2871  015154  005216                            INC     (SP)            ;SP HAS ODD BYTE ADR. OF RHBA
2872  015156  112736  000377                    MOVB    #377,   @(SP)+  ;LOAD ODD BYTE OF RHBA
2873  015162  011437  001126                    MOV     @R4,@#$BDDAT    ;TEST DATA
2874  015166  022737  177652  001126            CMP     #177652,@#$BDDAT;COMPARE MOVB RESULTS
2875  015174  001406                            BEQ     1$              ;BRANCH IF GOOD
2876  015176  012737  177652  001124            MOV     #177652,@#$GDDAT;GOOD DATA
2877  015204  010437  040172                    MOV     R4,     @#REGADR ;FAILING REGISTER RHBA
2878  015210  104001                            ERROR   1               ;MOVING INTO ODD BYTE OF
2879                                                                    ;RHBA GAVE WRONG RESULTS
2880  015212  112714  000125            1$:     MOVB    #125,   @R4
2881  015216  011437  001126                    MOV     @R4,@#$BDDAT    ;TEST DATA
2882  015222  022737  177524  001126            CMP     #177524,@#$BDDAT
2883  015230  001406                            BEQ     TST40           ;;BRANCH IF GOOD
2884  015232  012737  177524  001124            MOV     #177524,@#$GDDAT;GOOD DATA
2885  015240  010437  040172                    MOV     R4,     @#REGADR ;FAILING REGISTER RHBA
2886  015244  104001                            ERROR   1               ;MOVING INTO EVEN BYTE OF
2887                                                                    ;RHBA GAVE WRONG RESULTS
```

```
2888
2889                                    .SBTTL  DCL COMMAND TESTS
2890
2891                                    ;*****************************************************************
2892                                    ;*      FOUR GENERAL REGISTERS WILL BE RESERVED FOR HARDWARE
2893                                    ;*      R1=RHCS1 CONTROL AND STATUS1
2894                                    ;*      R2=RHCS2 CONTROL AND STATUS2
2895                                    ;*      R3=RHDS1 DRIVE STATUS 1
2896                                    ;*      R4=RHER1 ERROR REGISTER1
2897
2898                                    ;*      WHENEVER ANY OTHER USE IS MADE OF THESE REGISTERS
2899                                    ;*      APPROPIATE SAVING MUST BE DONE
2900
2901                                    ;*****************************************************************
2902
2903
2904
2905
2906
2907
2908                                    ;*****************************************************************
2909                                    ;*      ERROR REGISTER #01 (RHER1) TEST
2910                                    ;*      BIT #1 (ILLEGAL REGISTER) CANNOT BE TESTED ON PDP11 THIS BIT
2911                                    ;*      IS FOR PDP10 USE ONLY
2912                                    ;*****************************************************************
2913
2914
2915
2916                                    ;*****************************************************************
2917                                    ;*TEST 40       TEST ILF BIT #0 IN REG. RHER1
2918
2919                                    ;*      ALL 3 ILLEGAL FUNCTION CODES SHOULD SET - ATA,ERR,ILF - AND ARE TESTED
2920                                    ;*      A GO WITHOUT CLEARING ILF ERR SHOULD SET - MXF,DLT,TRE - BITS AND THEY ARE ALSO
2921
2922                                    ;*****************************************************************
2923  015246  000004                    TST40:  SCOPE
2924
2925  015250  012706  001000                    MOV     #STACK,SP       ;RESET STACK
2926  015254  012737  000040  002020            MOV     #TTNO,@#TSTNM   ;THIS SAVES TEST NUMBER
2927  015262  004737  040522                    JSR     PC,@#CLDISK     ;CLEAR REGISTERS
2928  015266  012777  000001  164356            MOV     #DMD,@RHMR      ;SET DIAGNOSTIC MODE
2929  015274  005037  002010                    CLR     @#TMPILL        ;GET READY TO MAKE ILLEGAL FUNCTION
2930  015300  012700  002024          1$:       MOV     #FUTABL,R0      ;LOAD FUNCTION CODE TABLE START
2931  015304  012705  000021                    MOV     #17.,R5         ;COUNTER (16 GOOD FUNCTIONS)
2932  015310  023720  002010          2$:       CMP     @#TMPILL,(R0)+  ;IS THIS A LEGAL FUNCTION CODE?
2933  015314  001004                    BNE     3$              ;NO - DECR. FUNCT. CODE CTR
2934  015316  062737  000002  002010            ADD     #2,@#TMPILL     ;YES MAKE NEXT FUNCTION CODE
2935  015324  000765                    BR      1$              ;TEST NEXT FUNCTION CODE
2936  015326  005305                    3$:     DEC     R5              ;MAKE NEXT CODE IF 1ST 16
2937                                                                    ;LEGAL FUNCTIONS NOT DONE
2938  015330  001367                    BNE     2$              ;BRANCH IF 16 NOT COMPLETE
2939  015332  032737  000100  002010            BIT     #100,@#TMPILL   ;ALL BITS UP TO BIT #5 COMPARED?
2940  015340  001077                    BNE     12$             ;YES - EXIT ------->
2941  015342  013737  002010  002066            MOV     @#TMPILL,@#ILLEGL;NO - TEST THE ILLEGAL FUNCTION
```

```
2942  015350  062737  000002  002010          ADD     #2,@#TMPILL     ;TEST NEW FUNCTION CODE NEXT TIME
2943
2944  015356  004737  040522          4$:     JSR     PC,@#CLDISK
2945  015362  012777  000001  164262          MOV     #DMD,@RHMR      ;SET DIAGNOSTIC MODE
2946  015370  013711  002066                  MOV     @#ILLEGL,@R1    ;ILLEGAL FUNCTION ---> RHCS1
2947  015374  012737  015356  001110          MOV     #4$,@#SLPERR    ;ERROR RETURN POINT
2948  015402  004737  040126                  JSR     PC,@#PUTREG     ;SAVE REGISTERS
2949  015406  005737  001704                  TST     @#ER1           ;THERE SHOULD NOT BE ANY ERROR YET
2950  015412  001403                          BEQ     5$              ;CONTINUE IF RHER1 STILL = 0
2951  015414  010437  001122                  MOV     R4,@#SBDADR     ;FAILING REGISTER ADDRESS RHER1
2952  015420  104011                          ERROR   11              ;ALTHOUGH AN ILLEGAL FUNCTION
2953                                                                  ;HAS BEEN MOVED INTO RHCS1
2954                                                                  ;NO ERRORS SHOULD SHOW TILL
2955                                                                  ;GO IS SET RHER1 SHOULD BE
2956                                                                  ;ALL ZEROS
2957
2958  015422  052711  000001          5$:     BIS     #GO,@R1         ;GO IN RHCS1
2959  015426  004737  040126                  JSR     PC,@#PUTREG     ;SAVE REGISTERS
2960  015432  022737  000001  001704          CMP     #ILF,@#ER1      ;ILLEGAL FUNCTION BIT SHOULD BE SET
2961  015440  001403                          BEQ     6$              ;IT IS - CONTINUE
2962  015442  010437  001122                  MOV     R4,@#SBDADR     ;FAILING REGISTER ADDRESS RHER1
2963  015446  104011                          ERROR   11              ;ILLEGAL FUNCTION DID NOT
2964                                                                  ;SET ON AN ILLEGAL FUNCTION
2965                                                                  ;EXECUTION, THE ILLEGAL FUNCTION
2966                                                                  ;BEING EXECUTED IS IN RHCS1
2967
2968  015450  013746  001724          6$:     MOV     @#DS1,-(SP)     ;GET RHDS1
2969  015454  042716  001000                  BIC     #PROG,(SP)      ;MASK PROG
2970  015460  022726  140700                  CMP     #ATA!ERR!VV!DPR!DRY,(SP)+
2971                                                                  ;ATTENTION (BIT 15)
2972                                                                  ;VOLUME VALID (BIT 6)
2973                                                                  ;COMPOSIT ERROR (BIT 14)
2974                                                                  ;DEVICE READY (BIT 7) SHOULD
2975                                                                  ;BE SET ON RHDS1
2976  015464  001404                          BEQ     7$              ;THEY ARE - CONTINUE
2977  015466  013737  001654  001122          MOV     @#RHDS1,@#SBDADR ;FAILING REGISTER ADDRESS RHDS1
2978  015474  104011                          ERROR   11              ;FOLLOWING BITS SHOULD BE SET
2979                                                                  ;WITH AN ILLEGAL FUNCTION
2980                                                                  ;ATTENTION (BIT 15)
2981                                                                  ;COMPOSIT ERROR (BIT 14)
2982                                                                  ;MEDIUM ON LINE (BIT 12)
2983                                                                  ;DEVICE READY (BIT 7)
2984
2985  015476  004737  042762          7$:     JSR     PC,@#MIDDLE     ;GIVE A WRITE HEADER AND
2986                                                                  ;DATA COMMAND WITHOUT
2987                                                                  ;CLEARING THE ERRORS
2988                                                                  ;USING "MIDDLE" SO THAT
2989                                                                  ;IT WILL COME BACK BEFORE
2990                                                                  ;THE END TO FIND OUT ITS
2991                                                                  ;STATE
2992  015502  010237  015510                  MOV     R2,@#10$        ;MOVE RHCS2 ADDRESS
2993  015506  104424                          WAT                     ;WAIT FOR 'MXF' BIT
2994  015510  000000          10$:    .WORD   0                       ;ADDRESS OF RHCS2
2995  015512  001000                  MXF
```

```
2996  015514  004737  040126          11$:    JSR     PC,@#PUTREG     ;SAVE REGISTERS
2997
2998  015520  032737  040000  001702          BIT     #TRE,@#CS1      ;TRANSFER ERROR (BIT 14) RHCS1 - 'TRE'
2999                                                                  ;SHOULD SET DUE TO 'MXF'
3000  015526  001003                          BNE     13$             ;IT IS - CONTINUE
3001  015530  010137  001122                  MOV     R1,@#$BDADR     ;FAILING REGISTER RHCS1
3002  015534  104011                          ERROR   11              ;TRANSFER ERROR (BIT 14) RHCS1 - 'TRE'
3003                                                                  ;SHOULD BE SET DUE TO 'MXF'
3004                                                                  ;LOCAL SCOPE RETURN POINT
3005
3006  015536  000660                  13$:    BR      1$              ;GO BACK & TEST NEXT FUNCTION CODE
3007
3008  015540  000240                  12$:    NOP
3009
```

```
3010
3011
3012                                    ;*****************************************************
3013                                    ;*TEST 41       READ IN PRESET
3014
3015                                    ;*      ALL POSSIBLE REGISTERS WILL BE FILLED WITH ONES
3016                                    ;*      THE REGISTER CONTENTS WILL BE SAVED IN REINTO BUFFER
3017                                    ;*      THE READ IN PRESET COMMAND WILL BE GIVEN
3018                                    ;*      ALL REGISTERS WILL BE CHECKED
3019
3020                                    ;*************************************************************
3021  015542  000004                    TST41:  SCOPE
3022  015544  012706  001000                    MOV     #STACK,SP       ;RESET STACK
3023  015550  012737  000041  002020            MOV     #TTNO,@#TSTNM   ;THIS SAVES TEST NUMBER
3024
3025  015556  004737  040522                    JSR     PC,@#CLDISK     ;INIT AND SET GENERAL REGISTERS
3026
3027                                            ;FILL ALL POSSIBLE BITS WITH ONES
3028
3029  015562  012777  177777  164034           MOV     #177777,@RHWC   ;WORD COUNT REGISTER GETS 177777
3030  015570  012777  177777  164030           MOV     #177777,@RHBA   ;BUS ADDRESS REGISTER GETS 177777
3031  015576  012777  017437  164032           MOV     #17437,@RHDST   ;DESIRED SECTOR TRACK GETS 17437
3032  015604  012777  016377  164030           MOV     #16377,@RHOF    ;OFFSET REGISTER GETS 16277
3033  015612  012777  000777  164024           MOV     #777,@RHCA      ;DESIRED CYLINDER GETS 777
3034  015620  012746  001400                    MOV     #A16!A17,-(SP)  ;GET BIT 9 AND 8
3035  015624  053716  002064                    BIS     @#READIN,(SP)
3036  015630  012677  163776                    MOV     (SP)+,@RHCS1    ;FILL READ IN PRESET IN RHCS1
3037  015634  012777  000001  164010           MOV     #DMD,@RHMR      ;SET DIAGNOSTIC MODE
3038
3039                                            ;THE REGISTERS WILL BE SAVED IN REINTO BUFFER
3040  015642  004037  041214                    JSR     R0,@#SAVER      ;SAVE
3041  015646  001624                            RHWC                    ;FROM
3042  015650  003134                            REINTO                  ;TO
3043  015652  000021                            17.                     ;NUMBER SAVED
3044
3045                                            ;GIVE READ IN PRESET COMMAND
3046  015654  052777  000001  163750           BIS     #GO,@RHCS1      ;INCLUDE GO TO READ IN PRESET
3047
3048                                            ;NOW SAVED REGISTERS WILL BE CHANGED TO EXPECTED VALUE
3049  015662  005037  003146                    CLR     @#REINTO+12     ;CLEAR SAVED RHDST
3050  015666  042737  016000  003152           BIC     #FMT22!HCI!ECI,@#REINTO+16 ;CLEAR FMT22,HCI,ECI IN
3051                                                                    ;SAVED RHOF
3052  015674  052737  000100  003152           BIS     #VV,@#REINTO+16 ;SET VV IN SAVED RHOF
3053  015702  005037  003154                    CLR     @#REINTO+20     ;CLEAR SAVED RHCA
3054
3055                                            ;AFTER A READ IN PRESET COMMAND
3056                                            ;SAVE REGISTERS AGAIN SO THAT COMPARES CAN BE DONE
3057
3058  015706  004037  041214                    JSR     R0,@#SAVER      ;SAVE
3059  015712  001624                            RHWC                    ;FROM
3060  015714  002070                            WRFROM                  ;TO
3061  015716  000021                            17.                     ;NUMBER OF REGISTERS SAVED
3062
3063                                            ;AS UPPER BYTE OF RHAS CAN BE CHANGING IN A DUAL PORT
```

```
3064                                            ;OPERATION THE UPPER BYTE OF RHAS WILL BE SAVED AS IS
3065                                            ;SO THAT THE COMPARES ARE ONLY VALID FOR THE LOWER BYTE
3066  015720  113737  003161  002115            MOVB    @#REINTO+25,@#WRFROM+25;SAVE UPPER RHAS
3067
3068                                            ;COMPARE REGISTERS BEFORE READ IN PRESET COMMAND
3069                                            ;WITH AFTER COMMAND
3070
3071  015726  004037  041416                    JSR     R0,@#COMPAR     ;COMPARE
3072  015732  003134                            REINTO                  ;GOOD BUFFER
3073  015734  002070                            WRFROM                  ;TEST BUFFER
3074  015736  000021                            17.                     ;NUMBER OF REGISTERS
3075  015740  015746                            1$                      ;RETURN FOR ERROR
3076  015742  015746                            1$                      ;SAME
3077  015744  015766                            2$                      ;RETURN FOR GOOD COMPARISON
3078
3079  015746  013705  045272          1$:       MOV     @#ERWORD,R5     ;GETTING READY TO INDEX
3080  015752  060505                            ADD     R5,R5           ;DOUBLE ERROR WORD
3081  015754  016537  001622  040172            MOV     RH%C-2(R5),@#REGADR ;FAILING REG. ADDRESS
3082  015762  104001                            ERROR   1               ;READ IN PRESET CAUSED IMPROPER
3083                                                                    ;REGISTER CHANGE
3084  015764  000207                            RTS     PC              ;RETURN FOR FURTHER COMPARISONS
3085
3086  015766                          2$:                               ;NO ERRORS
3087
```

```
                                        ;*****************************************************************
                                        ;*TEST 42       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
                                        ;*****************************************************************
016766  000004                    TST42:  SCOPE
016770  012737  000042  002020            MOV     #TTNO,@#TSTNM   ;THIS SAVES TEST NUMBER

                                        ;START WITH CLR IN RHCS2 (BIT5)
016776  004737  040522                    JSR     PC,@#CLDISK     ;CLEAR ALL POSSIBLE BITS
017002  012777  000001  163642            MOV     #DMD,@RHMR      ;SET DIAGNOSTIC MODE
017010  013711  002024                    MOV     @#NOPERA,@R1    ;PUT NOP OPERATION=0 IN RHCS1
017014  012700  001624                    MOV     #RHWC,R0        ;STARTING ADDRESS OF REG
017020  012703  001674                    MOV     #WC,R3          ;STARTING ADDRESS OF WHERE SAVED
017024  012702  000021                    MOV     #RHEC2-RHWC+2/2,R2 ;NUMBER OF REGISTERS
017030  013023                    1$:     MOV     @(R0)+,(R3)+    ;SAVE HARDWARE REG
017032  005302                            DEC     R2              ;COUNT
017034  001375                            BNE     1$              ;BRANCH IF NOT COMPLETE
017036  013737  001654  016056            MOV     @#RHDS1,@#2$    ;GET ADDRESS OF DRIVE STATUS
017044  010137  016064                    MOV     R1,@#3$         ;GET ADDRESS OF RHCS1
017050  052711  000001                    BIS     #GO,@R1         ;GO TO RHCS1
017054  104424                            WAT                     ;WAIT FOR DRY IN RHDS1
017056  000000                    2$:     .WORD   0               ;ADDRESS OF DRIVE STATUS RHDS1
017060  000200                            DRY                     ;DRY WILL BE WAITED ON
017062  104424                            WAT                     ;WAIT FOR RDY IN RHCS1
017064  000000                    3$:     .WORD   0               ;ADDRESS OF RHCS1 PUT HERE BY AN
                                                                  ;EARLIER MOV
017066  000200                            RDY                     ;RDY WILL BE WAITED ON

                                        ;AFTER A NO OP  COMMAND
                                        ;SAVE REGISTERS AGAIN SO THAT COMPARES CAN BE DONE

017070  004037  041214                    JSR     R0,@#SAVER      ;SAVE
017074  001624                            RHWC                    ;FROM
017076  002070                            WRFROM                  ;TO
017100  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
017102  113737  001721  002115            MOVB    @#AS+1,@#WRFROM+25;SAVE UPPER RHAS

                                        ;COMPARE REGISTERS BEFORE  NO OP COMMAND
                                        ;WITH AFTER COMMAND

017110  004037  041416                    JSR     R0,@#COMPAR     ;COMPARE
017114  001674                            WC                      ;GOOD BUFFER
017116  002070                            WRFROM                  ;TEST BUFFER
```

```
3142  016120  000021                          17.                     ;NUMBER OF REGISTERS
3143  016122  016130                          4$                      ;RETURN FOR ERROR
3144  016124  016130                          4$                      ;SAME
3145  016126  016150                          5$                      ;RETURN FOR GOOD COMPARISON
3146
3147  016130  013705  045272          4$:     MOV     @#ERWORD,R5     ;GETTING READY TO INDEX
3148  016134  060505                          ADD     R5,R5           ;DOUBLE ERROR WORD
3149  016136  016537  001622  040172          MOV     RHWC-2(R5),@#REGADR ;FAILING REG. ADDRESS
3150  016144  104001                          ERROR   1               ;NO OP COMMAND CAUSED IMPROPER
3151                                                                  ;REGISTER CHANGE
3152  016146  000207                          RTS     PC              ;RETURN FOR FURTHER COMPARISONS
3153
3154  016150                          5$:                             ;NO ERRORS
3155
3156
3157
3158  016150  012737  016156  001110          MOV     #14$,@#SLPERR   ;SET SCOPE LOOP TO 14$
3159  016156  004737  040522          14$:    JSR     PC,@#CLDISK     ;INIT LAST ALL ZERO TEST
3160  016162  012777  000001  163462          MOV     #DMD,@RHMR      ;SET DIAGNOSTIC MODE
3161
3162
3163                                          ;NOW START WITH ALL ONES IN ALL POSSIBLE REGISTERS
3164
3165  016170  012700  001624                  MOV     #RHWC,R0        ;ADDRESS OF FIRST REGISTER
3166  016174  012705  000021                  MOV     #RHEC2-RHWC+2/2,R5 ;NO. OF REGISTERS
3167  016200  012730  177676          6$:     MOV     #177676,@(R0)+  ;FILL WITH ALL ONES
3168  016204  013777  001762  163416          MOV     @#UNIT,@RHCS2   ;REINSTATE UNIT NUMBER UNDER TEST
3169                                                                  ;KEEP INTERRUPT DISABLED
3170  016212  005305                          DEC     R5              ;COUNT
3171  016214  001371                          BNE     6$              ;BRANCH IF INCOMPLETE
3172  016216  013711  002024                  MOV     @#NOPERA,@R1    ;PUT NOP OPERATION =0 IN RHCS1
3173  016222  012700  001624                  MOV     #RHWC,R0        ;STARTING ADDRESS OF REG
3174  016226  012703  001674                  MOV     #WC,R3          ;STARTING ADDRESS OF WHERE SAVED
3175  016232  012702  000021                  MOV     #RHEC2-RHWC+2/2,R2 ;NUMBER OF REGISTERS
3176  016236  013023                  7$:     MOV     @(R0)+,(R3)+    ;SAVE HARDWARE REG
3177  016240  005302                          DEC     R2              ;COUNT
3178  016242  001375                          BNE     7$              ;BRANCH IF NOT COMPLETE
3179  016244  013737  001654  016264          MOV     @#RHDS1,@#10$   ;GET ADDRESS OF DRIVE STATUS
3180  016252  010137  016272                  MOV     R1,@#11$        ;GET ACDRESS OF RHCS1
3181  016256  052711  000001                  BIS     #GO,@R1         ;GO TO RHCS1
3182  016262  104424                          WAT                     ;WAIT FOR DRY IN RHDS1
3183  016264  000000                  10$:    .WORD   0               ;ADDRESS OF DRIVE STATUS RHDS1
3184  016266  000200                          DRY                     ;DRY WILL BE WAITED ON
3185  016270  104424                          WAT                     ;WAIT FOR RDY IN RHCS1
3186  016272  000000                  11$:    .WORD   0               ;ADDRESS OF RHCS1 PUT HERE BY AN
3187                                                                  ;EARLIER MOV.
3188  016274  000200                          RDY                     ;RDY WILL BE WAITED ON
3189
3190                                          ;AFTER A NO OP COMMAND
3191                                          ;SAVE REGISTERS AGAIN SO THAT COMPARES CAN BE DONE
3192
3193  016276  004037  041214                  JSR     R0,@#SAVER      ;SAVE
3194  016302  001624                          RHWC                    ;FROM
3195  016304  002070                          WRFROM                  ;TO
```

```
3196  016306  000021                          17.                     ;NUMBER OF REGISTERS SAVED
3197
3198                                          ;AS UPPER BYTE OF RHAS CAN BE CHANGING IN A DUAL PORT
3199                                          ;OPERATION THE UPPER BYTE OF RHAS WILL BE SAVED AS IS
3200                                          ;SO THAT THE COMPARES ARE ONLY VALID FOR THE LOWER BYTE
3201  016310  113737  001721  002115          MOVB    @#AS+1,@#WRFROM+25;SAVE UPPER RHAS
3202
3203
3204                                          ;COMPARE REGISTERS BEFORE NO OP COMMAND
3205                                          ;WITH AFTER COMMAND
3206
3207  016316  004037  041416                  JSR     R0,@#COMPAR     ;COMPARE
3208  016322  001674                          WC                      ;GOOD BUFFER
3209  016324  002070                          WRFROM                  ;TEST BUFFER
3210  016326  000021                          17.                     ;NUMBER OF REGISTERS
3211  016330  016336                          12$                     ;RETURN FOR ERROR
3212  016332  016336                          12$                     ;SAME
3213  016334  016356                          13$                     ;RETURN FOR GOOD COMPARISON
3214
3215  016336  013705  045272          12$:    MOV     @#ERWORD,R5     ;GETTING READY TO INDEX
3216  016342  060505                          ADD     R5,R5           ;DOUBLE ERROR WORD
3217  016344  016537  001622  040172          MOV     RH#C-2(R5),@#REGADR ;FAILING REG. ADDRESS
3218  016352  104001                          ERROR   1               ;NO OP COMMAND CAUSED IMPROPER
3219                                                                  ;REGISTER CHANGE
3220  016354  000207                          RTS     PC              ;RETURN FOR FURTHER COMPARISONS
3221
3222  016356                          13$:                            ;NO ERRORS
```

```
                                        ;*****************************************************************
                                        ;*TEST 43        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

                                        ;*****************************************************************
016356  000004                  TST43:  SCOPE
016360  012706  001000                  MOV     #STACK,SP       ;RESET STACK
016364  012737  000043  002020          MOV     #TTNO,@#TSTNM   ;THIS SAVES TEST NUMBER
016372  004737  040522                  JSR     PC,@#CLDISK     ;SET REGISTERS AND CLEAR

                                        ;FILL ALL POSSIBLE BITS WITH ONES

016376  012777  177777  163216          MOV     #177777,@RHDB   ;BUS ADDRESS REGISTER GETS 177777
016404  012777  177777  163212          MOV     #177777,@RHWC   ;WORD COUNT REGISTER GETS 177777
016412  012777  177777  163206          MOV     #177777,@RHBA   ;BUS ADDRESS REGISTER GETS 177777
016420  052777  157010  163202          BIS     #157010,@RHCS2  ;CONTROL AND STATUS 2 GETS 157010
016426  012777  001476  163176          MOV     #1476,@RHCS1    ;CONTROL AND STATUS REGISTER/GETS 1476
016434  012777  177777  163172          MOV     #177777,@RHER1  ;ERROR REGISTER1 GETS 177777
016442  012777  017437  163166          MOV     #17437,@RHDST   ;DESIRED SECTOR TRACK
016450  012777  177777  163162          MOV     #177777,@RHER2  ;ERROR REGISTER 2
016456  012777  016277  163156          MOV     #16277,@RHOF    ;OFFSET REGISTER
016464  012777  177777  163152          MOV     #177777,@RHCA   ;DESIRED CYLINDER
016472  012777  177777  163146          MOV     #177777,@RHER3  ;ERROR REGISTER 3
016500  012777  000001  163144          MOV     #DMD,@RHMR      ;MAINTENANCE REGISTER
016506  012777  177777  163136          MOV     #177777,@RHMR   ;MAINTENANCE REGISTER

                                        ;THIS SETS BITS FOR ALL PRESENT DRIVES

016514  013700  002006                  MOV     @#TOTALAT,R0    ;GET DRIVE PRESENT
016520  005012                          CLR     @R2             ;CLEAR RHCS2 AND CARRY BIT
016522  012705  000010                  MOV     #8.,R5          ;COUNTER
016526  006000                  30$:    ROR     R0              ;GET BIT INTO CARRY
016530  103002                          BCC     31$             ;BRANCH IF NO UNIT ON THIS BIT
016532  012714  177777                  MOV     #-1,@R4         ;MOVE INTO ERROR REGISTER TO SET ATA
016536  005212                  31$:    INC     @R2             ;INCREMENT RHCS2 - UNIT NO.
016540  005305                          DEC     R5              ;COUNT
016542  001401                          BEQ     27$             ;BRANCH IF 8 DONE
016544  000770                          BR      30$             ;CONTINUE THIS ROUTINE
016546  013746  001762          27$:    MOV     @#UNIT,-(SP)
016552  052716  157010                  BIS     #157010,(SP)    ;REINSTATE SET BITS
016556  012612                          MOV     (SP)+,@R2       ;

016560  012777  000001  163064          MOV     #DMD,@RHMR      ;SET DMD
016566  013711  002032                  MOV     @#DCLEAR,@R1    ;DRIVE CLEAR = 10 INTO RHCS1
016572  052711  000001                  BIS     #GO,@R1         ;GO
```

```
3277  016576  012700  001622                    MOV     #RHDB,  R0      ;R0 CONTAINS ADDR. OF ADDR. OF REG.
3278
3279
3280                                            ;DATA BUFFER REGISTER
3281
3282
3283  016602  012737  177777  001124  2$:       MOV     #177777,        @#$GDDAT ;GOOD DATA FOR ERROR TYPEOUT
3284  016610  013037  001126                    MOV     @(R0)+,@#$BDDAT ;TEST DATA
3285  016614  022737  177777  001126            CMP     #177777,        @#$BDDAT;COMPARE DATA
3286  016622  001402                            BEQ     3$              ;BRANCH IF GOOD
3287  016624  004737  017462                    JSR     PC,@#ERCLFC     ;JUMP TO ERROR FOR CLR (BIT 5)
3288                                                                    ;IN RHCS2
3289
3290
3291                                            ;WORD COUNT REGISTER
3292
3293
3294  016630  012737  177777  001124  3$:       MOV     #177777,        @#$GDDAT ;GOOD DATA FOR ERROR TYPEOUT
3295  016636  013037  001126                    MOV     @(R0)+,@#$BDDAT ;TEST DATA
3296  016642  022737  177777  001126            CMP     #177777,        @#$BDDAT;COMPARE DATA
3297  016650  001402                            BEQ     4$              ;BRANCH IF GOOD
3298  016652  004737  017462                    JSR     PC,@#ERCLFC     ;JUMP TO ERROR FOR CLR (BIT 5)
3299                                                                    ;IN RHCS2
3300
3301
3302                                            ;BUS ADDRESS REGISTER
3303
3304
3305  016656  012737  177776  001124  4$:       MOV     #177776,        @#$GDDAT ;GOOD DATA FOR ERROR TYPEOUT
3306  016664  013037  001126                    MOV     @(R0)+,@#$BDDAT ;TEST DATA
3307  016670  022737  177776  001126            CMP     #177776,        @#$BDDAT;COMPARE DATA
3308  016676  001402                            BEQ     5$              ;BRANCH IF GOOD
3309  016700  004737  017462                    JSR     PC,@#ERCLFC     ;JUMP TO ERROR FOR CLR (BIT 5)
3310                                                                    ;IN RHCS2
3311
3312
3313                                            ;CONTROL AND STATUS 2 REGISTER
3314
3315
3316
3317  016704  012746  000110          5$:       MOV     #110,   -(SP)   ;INCLUDE IR
3318  016710  053716  001762                    BIS     @#UNIT, (SP)    ;SET UNIT NO.
3319  016714  012637  001124                    MOV     (SP)+,  @#$GDDAT;GOOD DATA FOR TYPE OUT
3320  016720  013037  001126                    MOV     @(R0)+,@#$BDDAT ;TEST DATA
3321  016724  023737  001124  001126            CMP     @#$GDDAT,@#$BDDAT;COMPARE DATA
3322  016732  001402                            BEQ     6$              ;BRANCH IF GOOD
3323  016734  004737  017462                    JSR     PC,@#ERCLFC     ;JUMP TO ERROR FOR CLR (BIT 5)
3324                                                                    ;IN RHCS2
3325
3326
3327                                            ;CONTROL AND STATUS 1 REGISTER
3328
3329
3330  016740  005737  001766          6$:       TST     @#NUNIT         ;ARE THERE MORE THAN ONE UNIT
```

```
3331  016744  001404                          BEQ     32$                 ;BRANCH IF ONLY ONE UNIT
3332  016746  012737  104210  001124          MOV     #104210,@#SGDDAT    ;GOOD DATA
3333  016754  000403                          BR      33$
3334  016756  012737  004210  001124  32$:    MOV     #4210,@#SGDDAT      ;GOOD DATA
3335  016764  013037  001126          33$:    MOV     @(R0)+,@#SBDDAT     ;TEST DATA
3336
3337  016770  023737  001124  001126          CMP     @#SGDDAT,@#SBDDAT   ;COMPARE DATA
3338  016776  001402                          BEQ     7$                  ;BRANCH IF GOOD
3339  017000  004737  017462                  JSR     PC,@#ERCLFC         ;JUMP TO ERROR FOR CLR BIT 5
3340                                                                      ;IN RHCS2
3341
3342                                          ;ERROR 1 REGISTER
3343
3344
3345  017004  012737  000000  001124  7$:     MOV     #0,     @#SGDDAT    ;GOOD DATA FOR ERROR TYPEOUT
3346  017012  013037  001126                  MOV     @(R0)+,@#SBDDAT     ;TEST DATA
3347  017016  022737  000000  001126          CMP     #0,     @#SBDDAT    ;COMPARE DATA
3348  017024  001402                          BEQ     10$                 ;BRANCH IF GOOD
3349  017026  004737  017462                  JSR     PC,@#ERCLFC         ;JUMP TO ERROR FOR CLR (BIT 5)
3350                                                                      ;IN RHCS2
3351
3352                                          ;DESIRED SECTOR/TRACK REGISTER
3353
3354
3355  017032  012737  017437  001124  10$:    MOV     #17437, @#SGDDAT    ;GOOD DATA FOR ERROR TYPEOUT
3356  017040  013037  001126                  MOV     @(R0)+,@#SBDDAT     ;TEST DATA
3357  017044  022737  017437  001126          CMP     #17437, @#SBDDAT    ;COMPARE DATA
3358  017052  001402                          BEQ     11$                 ;BRANCH IF GOOD
3359  017054  004737  017462                  JSR     PC,@#ERCLFC         ;JUMP TO ERROR FOR CLR (BIT 5)
3360                                                                      ;IN RHCS2
3361
3362                                          ;ERROR 2 REGISTER
3363
3364
3365  017060  012737  000000  001124  11$:    MOV     #0,     @#SGDDAT    ;GOOD DATA FOR ERROR TYPEOUT
3366  017066  013037  001126                  MOV     @(R0)+,@#SBDDAT     ;TEST DATA
3367  017072  022737  000000  001126          CMP     #0,     @#SBDDAT    ;COMPARE DATA
3368  017100  001402                          BEQ     12$                 ;BRANCH IF GOOD
3369  017102  004737  017462                  JSR     PC,@#ERCLFC         ;JUMP TO ERROR FOR CLR (BIT 5)
3370                                                                      ;IN RHCS2
3371
3372                                          ;OFFSET REGISTER
3373
3374
3375  017106  012737  116000  001124  12$:    MOV     #116000,        @#SGDDAT    ;GOOD DATA FOR ERROR TYPEOUT
3376  017114  013037  001126                  MOV     @(R0)+,@#SBDDAT     ;TEST DATA
3377  017120  022737  116000  001126          CMP     #116000,        @#SBDDAT;COMPARE DATA
3378  017126  001402                          BEQ     13$                 ;BRANCH IF GOOD
3379  017130  004737  017462                  JSR     PC,@#ERCLFC         ;JUMP TO ERROR FOR CLR (BIT 5)
3380                                                                      ;IN RHCS2
3381
3382                                          ;DESIRED CYLINDER ADDRESS REGISTER
3383
3384
```

```
3385  017134  012737  001777  001124  13$:    MOV     #1777,  @#$GDDAT ;GOOD DATA FOR ERROR TYPEOUT
3386  017142  013037  001126                  MOV     @(R0)+,@#$BDDAT ;TEST DATA
3387  017146  022737  001777  001126          CMP     #1777,  @#$BDDAT;COMPARE DATA
3388  017154  001402                          BEQ     14$             ;BRANCH IF GOOD
3389  017156  004737  017462                  JSR     PC,@#ERCLFC     ;JUMP TO ERROR FOR CLR (BIT 5)
3390                                                                  ;IN RHCS2
3391
3392                                          ;ERROR 3 REGISTER
3393
3394
3395  017162  012737  000000  001124  14$:    MOV     #0,     @#$GDDAT ;GOOD DATA FOR ERROR TYPEOUT
3396  017170  013037  001126                  MOV     @(R0)+,@#$BDDAT ;TEST DATA
3397  017174  022737  000000  001126          CMP     #0,     @#$BDDAT;COMPARE DATA
3398  017202  001402                          BEQ     15$             ;BRANCH IF GOOD
3399  017204  004737  017462                  JSR     PC,@#ERCLFC     ;JUMP TO ERROR FOR CLR (BIT 5)
3400                                                                  ;IN RHCS2
3401
3402                                          ;ATTENTION SUMMARY REGISTER
3403
3404  017210  013737  002006  001124  15$:    MOV     @#TOTALAT,@#$GDDAT;SET ALL BITS OF DRIVE PRESENT IN RHAS
3405  017216  043737  002004  001124          BIC     @#ATTENT,@#$GDDAT       ;CLEAR ONLY WORKING DRIVE BIT
3406  017224  013037  001126                  MOV     @(R0)+,@#$BDDAT         ;GET RHAS
3407  017230  123737  001124  001126          CMPB    @#$GDDAT,@#$BDDAT       ;COMPARE DATA
3408  017236  001402                          BEQ     16$                     ;BRANCH IF GOOD
3409  017240  004737  017462                  JSR     PC,@#ERCLFC             ;JUMP TO ERROR FOR CLR (BIT 5) IN RHCS2
3410
3411                                          ;MAINTAINABILITY REGISTER
3412
3413
3414  017244  012737  000400  001124  16$:    MOV     #400,   @#$GDDAT ;GOOD DATA FOR ERROR TYPEOUT
3415  017252  013037  001126                  MOV     @(R0)+,@#$BDDAT ;TEST DATA
3416  017256  022737  000400  001126          CMP     #400,   @#$BDDAT;COMPARE DATA
3417  017264  001402                          BEQ     17$             ;BRANCH IF GOOD
3418  017266  004737  017462                  JSR     PC,@#ERCLFC     ;JUMP TO ERROR FOR CLR (BIT 5)
3419                                                                  ;IN RHCS2
3420
3421                                          ;DRIVE STATUS REGISTER
3422
3423  017272  012737  000700  001124  17$:    MOV     #700,@#$GDDAT   ;GOOD DATA FOR PRINTOUT
3424  017300  013046                          MOV     @(R0)+,-(SP)    ;GET RHDS1
3425  017302  011637  001126                  MOV     (SP),@#$BDDAT   ;TEST DATA
3426  017306  042716  001000                  BIC     #PROG,(SP)      ;CLEAR PROG BIT
3427  017312  022726  000700                  CMP     #700,(SP)+      ;COMPARE DATA
3428  017316  001402                          BEQ     20$             ;BRANCH IF GOOD
3429  017320  004737  017462                  JSR     PC,@#ERCLFC     ;JUMP TO ERROR FOR DRIVE CLEAR
3430
3431                                          ;DRIVE TYPE
3432
3433
3434  017324  013737  001776  001124  20$:    MOV     @#SAVDT,        @#$GDDAT ;GOOD DATA FOR ERROR TYPEOUT
3435  017332  013037  001126                  MOV     @(R0)+,@#$BDDAT ;TEST DATA
3436  017336  023737  001776  001126          CMP     @#SAVDT,        @#$BDDAT;COMPARE DATA
3437  017344  001402                          BEQ     21$             ;BRANCH IF GOOD
3438  017346  004737  017462                  JSR     PC,@#ERCLFC     ;JUMP TO ERROR FOR CLR (BIT 5)
```

```
3439                                                        ;IN RHCS2
3440
3441                                    ;SERIAL NUMBER REGISTER
3442
3443
3444
3445  017352  013737  002000  001124  21$:  MOV   @#SAVSN,       @#$GDDAT ;GOOD DATA FOR ERROR TYPEOUT
3446  017360  013037  001126              MOV   @(R0)+,@#$BDDAT ;TEST DATA
3447  017364  023737  002000  001126      CMP   @#SAVSN,       @#$BDDAT;COMPARE DATA
3448  017372  001402                      BEQ   22$             ;BRANCH IF GOOD
3449  017374  004737  017462              JSR   PC,@#ERCLFC     ;JUMP TO ERROR FOR CLR (BIT 5)
3450                                                            ;IN RHCS2
3451
3452                                    ;ECC1 POSITION
3453
3454
3455  017400  012737  000000  001124  22$:  MOV   #0,     @#$GDDAT ;GOOD DATA FOR ERROR TYPEOUT
3456  017406  013037  001126              MOV   @(R0)+,@#$BDDAT ;TEST DATA
3457  017412  022737  000000  001126      CMP   #0,     @#$BDDAT;COMPARE DATA
3458  017420  001402                      BEQ   23$             ;BRANCH IF GOOD
3459  017422  004737  017462              JSR   PC,@#ERCLFC     ;JUMP TO ERROR FOR CLR (BIT 5)
3460                                                            ;IN RHCS2
3461
3462                                    ;ECC2 PATTERN
3463
3464
3465  017426  012737  000000  001124  23$:  MOV   #0,     @#$GDDAT ;GOOD DATA FOR ERROR TYPEOUT
3466  017434  013037  001126              MOV   @(R0)+,@#$BDDAT ;TEST DATA
3467  017440  022737  000000  001126      CMP   #0,     @#$BDDAT;COMPARE DATA
3468  017446  001402                      BEQ   24$             ;BRANCH IF GOOD
3469  017450  004737  017462              JSR   PC,@#ERCLFC     ;JUMP TO ERROR FOR CLR (BIT 5)
3470                                                            ;IN RHCS2
3471
3472
3473                                    ;LOOK-AHEAD REGISTER
3474
3475  017454  005720              24$:  TST   (R0)+             ;AS THE LOOK-AHEAD REG. CANNOT BE PREDICTED
3476                                                            ;IT IS NOT CHECKED AFTER AN INIT
3477
3478                                    ;CURRENT CYLINDER ADDRESS REGISTER
3479
3480  017456  005720              25$:  TST   (R0)+             ;AS THE CURRENT CYL REG. CANNOT BE PREDICTED
3481                                                            ;AFTER AN INIT IT IS NOT CHECKED
3482
3483  017460                      26$:
3484  017460  000413                      BR    TST44           ;BRANCH OVER JSR
3485
3486  017462  014037  040172  ERCLFC: MOV   -(R0),  @#REGADR ;FAILING REGISTER ADDRESS
3487  017466  104001                      ERROR 1               ;CLR FUNCTION = 10 IN RHCS1 DID
3488                                                            ;NOT CLEAR APPROPIATE BITS
3489                                                            ;OR CLEARED EXTRA BITS
3490  017470  005720                      TST   (R0)+           ;UNDO -(R0) FOR ERROR
3491  017472  000207                      RTS   PC              ;RETURN TO ABOVE PROGRAM
3492                                                            ;OR CLEARED EXTRA BITS
```

```
3493  017474  005720                          TST     (R0)+           ;UNDO -(R0) FOR BAD DATA
3494  017476  004737  040556                  JSR     PC,@#CHECKT     ;CHECK THAT DVA,RDY,DPR,DRY = 1
3495  017502  104400  057176                  TYPE    ,CPHALT         ;AND THAT NO OTHERS = 1.  CANNOT CON-
3496                                                                  ;TINUE TESTING IF BOTH AREN'T TRUE
3497  017506  000000                          HALT                    ;STOP THE TEST
3498
```

```
3499
3500
3501
3502                                    ;*****************************************************************
3503                                    ;*TEST 44        SEEK COMMAND TEST
3504
3505                                    ;*      THE SEEK COMMAND WILL BE LOADED INTO RHCS1 WITH GO
3506                                    ;*      THEN ALL REGISTERS WILL BE CHECKED
3507                                    ;*      RH CLEAR WILL BE GIVEN
3508                                    ;*      THEN ALL REGISTERS WILL BE CHECKED
3509
3510                                    ;*****************************************************************
3511  017510  000004                   TST44:  SCOPE
3512  017512  012706  001000                   MOV     #STACK,SP       ;RESET STACK
3513  017516  012737  000044  002020           MOV     #TTNO,@#TSTNM   ;THIS SAVES TEST NUMBER
3514  017524  004737  040522                   JSR     PC,@#CLDISK     ;INIT AND SET UP GENERAL REG.
3515                                                                   ;AND UNIT NUMBER
3516
3517  017530  012777  000001  162114           MOV     #DMD,@RHMR      ;SET DIAGNOSTIC MODE BIT
3518                                                                   ;THIS ENABLES COMMANDS WITHOUT MOL
3519                                                                   ;AND HOLDS RHLA FROM MOVING
3520
3521
3522
3523  017536  005077  162074                   CLR     @RHDST          ;MAKE DESIRED SECTOR TRACK LEGAL
3524  017542  013777  002054  162062           MOV     @#SEECOM,@RHCS1 ;LOAD SEEK COMMAND INTO RH
3525  017550  017746  162114                   MOV     @RHCC,-(SP)     ;GET RHCC
3526  017554  022726  000632                   CMP     #410.,(SP)+     ;IS CURRENT CYLINDER SAME AS 410.
3527  017560  001404                           BEQ     9$              ;BRANCH IF YES TO MAKE RHCA =409.
3528  017562  012737  000632  001202           MOV     #410.,@#$TMP5   ;GET READY TO MAKE RHCA =410
3529  017570  000403                           BR      10$             ;BRANCH TO FILL RHCA
3530  017572  012737  000631  001202  9$:      MOV     #409.,@#$TMP5   ;GET READY TO MAKE RHCA =409.
3531  017600  013777  001202  162036  10$:     MOV     @#$TMP5,@RHCA   ;MAKE DESIRED CYLINDER 401. OR 409.
3532
3533                                    ;SAVE REGISTERS FOR COMPARISON AFTER GO
3534
3535  017606  004037  041214                   JSR     R0,@#SAVER      ;SAVE
3536  017612  001624                           RHWC                    ;FROM
3537  017614  003134                           REINTO                  ;TO
3538  017616  000023                           19.                     ;NUMBER OF REGISTERS SAVED
3539
3540                                    ;GIVE GO TO COMMAND
3541  017620  052777  000001  162004           BIS     #GO,@RHCS1      ;GO TO COMMAND
3542
3543                                    ;CHANGE SAVED REGISTERS TO EXPECTED VALUES
3544
3545  017626  052737  000001  003142           BIS     #GO,@#REINTO+6  ;SAVED RHCS1
3546  017634  052737  020000  003164           BIS     #PIP,@#REINTO+30 ;SAVED RHDS1
3547  017642  042737  000200  003164           BIC     #DRY,@#REINTO+30 ;SAVED RHDS1
3548
3549
3550                                    ;AFTER GO HAS BEEN GIVEN TO SEEK COMMAND
3551                                    ;SAVE REGISTERS AGAIN SO THAT COMPARISONS CAN
3552                                    ;BE DONE
```

```
3553
3554  017650  004037  041214                  JSR     R0,@#SAVER      ;SAVE
3555  017654  001624                          RHWC                    ;FROM
3556  017656  002070                          WRFROM                  ;TO
3557  017660  000023                          19.                     ;NUMBER OF REGISTERS SAVED
3558
3559                                          ;AS UPPER BYTE OF RHAS CAN BE CHANGING IN A DUAL PORT
3560                                          ;OPERATION THE UPPER BYTE OF RHAS WILL BE SAVED AS IS
3561                                          ;SO THAT THE COMPARES ARE ONLY VALID FOR THE LOWER BYTE
3562  017662  113737  003161  002115          MOVB    @#REINTO+25,@#WRFROM+25;SAVE UPPER RHAS
3563
3564                                          ;COMPARE REGISTERS BEFORE SEEK COMMAND
3565                                          ;WITH AFTER GO
3566
3567  017670  004037  041416                  JSR     R0,@#COMPAR     ;COMPARE
3568  017674  003134                          REINTO                  ;GOOD BUFFER
3569  017676  002070                          WRFROM                  ;TEST BUFFER
3570  017700  000023                          19.                     ;NUMBER
3571  017702  017710                          1$                      ;RETURN FOR ERROR
3572  017704  017710                          1$                      ;SAME
3573  017706  017730                          2$                      ;RETURN FOR GOOD COMPARISON
3574
3575  017710  013705  045272          1$:     MOV     @#ERWORD,R5     ;GETTING READY TO INDEX
3576  017714  060505                          ADD     R5,R5           ;DOUBLE ERROR WORD
3577  017716  016537  001622  040172          MOV     RHWC-2(R5),@#REGADR ;FAILING REGISTER ADDRESS
3578
3579  017724  104001                          ERROR   1               ;IMPROPER REGISTER CHANGE
3580                                                                  ;AFTER SEEK COMMAND
3581                                                                  ;WITH GO IS GIVEN
3582  017726  000207                          RTS     PC              ;RETURN TO COMPARISON
3583
3584
3585                                          ;NOW GIVE INIT AND GET GO AND PIP DOWN
3586
3587  017730  052712  000040          2$:     BIS     #CLR,@R2        ;RH INITILIZE
3588  017734  013712  001762                  MOV     @#UNIT,@R2      ;REINSTATE UNIT NUMBER
3589  017740  012777  000001  161704          MOV     #DMD,@RHMR      ;SET DIAGNOSTIC MODE BIT
3590                                                                  ;THIS ENABLES COMMANDS WITHOUT MOL
3591                                                                  ;AND HOLDS RHLA FROM MOVING
3592
3593                                          ;CHANGE REGISTERS TO EXPECTED VALUE
3594  017746  042737  000001  003142          BIC     #GO,@#REINTO+6  ;SAVED RHCS1
3595  017754  042737  020000  003164          BIC     #PIP,@#REINTO+30 ;SAVED RHDS1
3596  017762  052737  000200  003164          BIS     #DRY,@#REINTO+30 ;SAVED RHDS1
3597  017770  017737  161672  003176          MOV     @RHLA,@#REINTO+42;SAVED RHLA
3598  017776  013737  001202  003200          MOV     @#STMPS,@#REINTO+44 ;SAVED RHCC
3599
3600
3601                                          ;AFTER INITILIZE SAVE REGISTERS SO THAT
3602                                          ;COMPARES CAN BE DONE
3603
3604  020004  004037  041214                  JSR     R0,@#SAVER      ;SAVE
3605  020010  001624                          RHWC                    ;FROM
3606  020012  002070                          WRFROM                  ;TO
```

```
3607  020014  000023                          19.                     ;NUMBER OF REGISTERS SAVED
3608
3609
3610                                          ;AS UPPER BYTE OF RHAS CAN BE CHANGING IN A DUAL PORT
3611                                          ;OPERATION THE UPPER BYTE OF RHAS WILL BE SAVED AS IS
3612                                          ;SO THAT THE COMPARES ARE ONLY VALID FOR THE LOWER BYTE
3613  020016  113737  003161  002115          MOVB    @#REINTO+25,@#WRFROM+25;SAVE UPPER RHAS
3614
3615
3616                                          ;COMPARE REGISTERS AFTER INITILIZE
3617  020024  004037  041416                  JSR     R0,@#COMPAR     ;COMPARE
3618  020030  003134                          REINTO                  ;GOOD BUFFER
3619  020032  002070                          WRFROM                  ;TEST BUFFER
3620  020034  000023                          19.                     ;NUMBER OF REGISTERS TO BE
3621                                                                  ;COMPARED
3622  020036  020044                          3$                      ;RETURN POINT FOR ERROR
3623  020040  020044                          3$                      ;SAME
3624  020042  020064                          4$                      ;RETURN POINT FOR GOOD COMPARISON
3625
3626  020044  013705  045272          3$:     MOV     @#ERWORD,R5     ;GETTING READY TO INDEX
3627  020050  060505                          ADD     R5,R5           ;DOUBLE ERROR WORD
3628  020052  016537  001622  040172          MOV     RHMC-2(R5),@#REGADR ;FAILING REGISTER ADDRESS
3629  020060  104001                          ERROR   1               ;IMPROPER REGISTER
3630                                                                  ;CONTENTS AFTER GIVING AN
3631                                                                  ;INITILIZE FOLLOWING A
3632                                                                  ;SEEK COMMAND
3633  020062  000207                          RTS     PC              ;RETURN TO COMPARISON
3634
3635  020064                          4$:                             ;GOOD COMPARISON
3636
3637  020064  004737  040522                  JSR     PC,@#CLDISK     ;INIT AND SET UP GENERAL REG.
3638                                                                  ;AND UNIT NUMBER
3639  020070  012777  000001  161554          MOV     #DMD,@RHMR      ;SET DIAGNOSTIC MODE BIT
3640                                                                  ;THIS ENABLES COMMANDS WITHOUT MOL
3641                                                                  ;AND HOLDS RHLA FROM MOVING
3642
3643  020076  013777  002054  161526          MOV     @#SEECOM,@RHCS1 ;LOAD SEEK COMMAND INTO RHC
3644  020104  005077  161534                  CLR     @RHCA           ;DESIRED CYLINDER ADDRESS
3645                                          ;SAVE REGISTERS FOR COMPARISON AFTER GO
3646  020110  004037  041214                  JSR     R0,@#SAVER      ;SAVE
3647  020114  001624                          RHMC                    ;FROM
3648  020116  003134                          REINTO                  ;TO
3649  020120  000023                          19.                     ;NUMBER OF REGISTERS SAVED
3650
3651                                          ;GIVE GO TO SEEK COMMAND
3652  020122  052777  000001  161502          BIS     #GO,@RHCS1      ;GO TO SEEK COMMAND
3653
3654                                          ;CHANGE SAVED REGISTERS TO EXPECTED VALUES
3655
3656  020130  052737  000001  003142          BIS     #GO,@#REINTO+6  ;SAVED RHCS1
3657  020136  052737  020000  003164          BIS     #PIP,@#REINTO+30 ;SAVED RHDS1
3658  020144  042737  000200  003164          BIC     #DRY,@#REINTO+30 ;SAVED RHDS1
3659
3660
```

```
3661                                            ;AFTER GO HAS BEEN GIVEN TO SEEK COMMAND
3662                                            ;SAVE REGISTERS AGAIN SO THAT COMPARISONS CAN
3663                                            ;BE DONE
3664  020152  004037  041214                    JSR     R0,@#SAVER      ;SAVE
3665  020156  001624                            RHWC                    ;FROM
3666  020160  002070                            WRFROM                  ;TO
3667  020162  000023                            19.                     ;NUMBER OF REGISTERS SAVED
3668
3669
3670                                            ;AS UPPER BYTE OF RHAS CAN BE CHANGING IN A DUAL PORT
3671                                            ;OPERATION THE UPPER BYTE OF RHAS WILL BE SAVED AS IS
3672                                            ;SO THAT THE COMPARES ARE ONLY VALID FOR THE LOWER BYTE
3673  020164  113737  003161  002115            MOVB    @#REINTO+25,@#WRFROM+25;SAVE UPPER RHAS
3674
3675
3676                                            ;COMPARE REGISTERS BEFORE COMMAND
3677                                            ;WITH AFTER GO
3678
3679  020172  004037  041416                    JSR     R0,@#COMPAR     ;COMPARE
3680  020176  003134                            REINTO                  ;GOOD BUFFER
3681  020200  002070                            WRFROM                  ;TEST BUFFER
3682  020202  000023                            19.                     ;NUMBER
3683  020204  020212                            5$                      ;RETURN FOR ERROR
3684  020206  020212                            5$                      ;SAME
3685  020210  020232                            6$                      ;RETURN FOR GOOD COMPARISON
3686  020212  013705  045272          5$:       MOV     @#ERWORD,R5     ;GETTING READY TO INDEX
3687  020216  060505                            ADD     R5,R5           ;DOUBLG ERROR WORD
3688  020220  016537  001622  040172            MOV     RHWC-2(R5),@#REGADR ;FAILING REGISTER ADDRESS
3689  020226  104001                            ERROR   1               ;IMPROPER REGISTER CHANGE
3690                                                                    ;AFTER COMMAND
3691                                                                    ;WITH GO IS GIVEN
3692  020230  000207                            RTS     PC              ;RETURN TO COMPARISON
3693
3694                                            ;NOW GIVE INIT AND GET GO AND PIP DOWN
3695
3696  020232  052712  000040          6$:       BIS     #CLR,@R2        ;RH INITILIZE
3697  020236  013712  001762                    MOV     @#UNIT,@R2      ;REINSTATE UNIT NUMBER
3698  020242  012777  000001  161402            MOV     #DMD,@RHMR      ;SET DIAGNOSTIC MODE BIT
3699                                                                    ;THIS ENABLES COMMANDS WITHOUT MOL
3700                                                                    ;AND HOLDS RHLA FROM MOVING
3701
3702                                            ;CHANGE REGISTERS TO EXPECTED VALUE
3703
3704  020250  042737  000001  003142            BIC     #GO,@#REINTO+6  ;SAVED RHCS1
3705  020256  042737  020000  003164            BIC     #PIP,@#REINTO+30 ;SAVED RHDS1
3706  020264  052737  000200  003164            BIS     #DRY,@#REINTO+30 ;SAVED RHDS1
3707  020272  017737  161370  003176            MOV     @RHLA,@#REINTO+42;SAVED RHLA
3708  020300  005037  003200                    CLR     @#REINTO+44     ;SAVED RHCC
3709
3710
3711                                            ;AFTER INITIALIZE SAVE REGISTERS SO THAT
3712                                            ;COMPARES CAN BE DONE
3713
3714  020304  004037  041214                    JSR     R0,@#SAVER      ;SAVE
```

```
3715  020310  001624                          RHMC                    ;FROM
3716  020312  002070                          WRFROM                  ;TO
3717  020314  000023                          19.                     ;NUMBER OF REGISTERS SAVED
3718
3719                                          ;AS UPPER BYTE OF RHAS CAN BE CHANGING IN A DUAL PORT
3720                                          ;OPERATION THE UPPER BYTE OF RHAS WILL BE SAVED AS IS
3721                                          ;SO THAT THE COMPARES ARE ONLY VALID FOR THE LOWER BYTE
3722  020316  113737  003161  002115          MOVB    @#REINTO+25,@#WRFROM+25;SAVE UPPER RHAS
3723
3724
3725                                          ;COMPARE REGISTERS AFTER INITALIZE
3726
3727  020324  004037  041416                  JSR     R0,@#COMPAR     ;COMPARE
3728  020330  003134                          REINTO                  ;GOOD BUFFER
3729  020332  002070                          WRFROM                  ;TEST BUFFER
3730  020334  000023                          19.                     ;NUMBER OF REGISTERS TO BE
3731                                                                  ;COMPARED
3732  020336  020344                          7$                      ;RETURN POINT FOR ERROR
3733  020340  020344                          7$                      ;SAME
3734  020342  020364                          8$                      ;RETURN POINT FOR GOOD COMPARISON
3735
3736  020344  013705  045272          7$:     MOV     @#ERWORD,R5     ;GETTING READY TO INDEX
3737  020350  060505                          ADD     R5,R5           ;DOUBLE ERROR WORD
3738  020352  016537  001622  040172          MOV     RHMC-2(R5),@#REGADR ;FAILING REGISTER ADDRESS
3739  020360  104001                          ERROR   1               ;* CONTENTS AFTER GIVING AN
3740                                                                  ;  COMMAND
3741  020362  000207                          RTS     PC              ;RETURN TO COMPARISON
3742  020364                          8$:                             ;GOOD COMPARISON
```

```
3743
3744
3745                                    ;*****************************************************************
3746                                    ;*TEST 45       UNLOAD COMMAND TEST
3747
3748                                    ;*      THE UNLOAD COMMAND WILL BE LOADED INTO RHCS1 WITH GO
3749                                    ;*      THEN ALL REGISTERS WILL BE CHECKED
3750                                    ;*      RH CLEAR WILL BE GIVEN
3751
3752                                    ;*****************************************************************
3753  020364  000004                    TST45:  SCOPE
3754  020366  012706  001000                    MOV     #STACK,SP       ;RESET STACK
3755  020372  012737  000045  002020            MOV     #TTNO,@#TSTNM   ;THIS SAVES TEST NUMBER
3756  020400  004737  040522                    JSR     PC,@#CLDISK     ;INIT AND SET UP GENERAL REG.
3757                                                                    ;AND UNIT NUMBER
3758  020404  012777  000001  161240            MOV     #DMD,@RHMR      ;SET DIAGNOSTIC MODE BIT
3759                                                                    ;THIS ENABLES COMMANDS WITHOUT MOL
3760                                                                    ;AND HOLDS RHLA FROM MOVING
3761
3762  020412  013777  002026  161212            MOV     @#UNLOAD,@RHCS1 ;LOAD UNLOAD COMMAND INTO RH
3763
3764                                            ;SAVE REGISTERS FOR COMPARISON AFTER GO
3765
3766  020420  004037  041214                    JSR     R0,@#SAVER      ;SAVE
3767  020424  001624                            RHWC                    ;FROM
3768  020426  003134                            REINTO                  ;TO
3769  020430  000023                            19.                     ;NUMBER OF REGISTERS SAVED
3770
3771                                            ;GIVE GO TO UNLOAD COMMAND
3772  020432  052777  000001  161172            BIS     #GO,@RHCS1      ;GO TO UNLOAD COMMAND
3773
3774                                            ;CHANGE SAVED REGISTERS TO EXPECTED VALUES
3775  020440  052737  000001  003142            BIS     #GO,@#REINTO+6  ;SAVED RHCS1
3776  020446  052737  020000  003164            BIS     #PIP,@#REINTO+30 ;SAVED RHDS1
3777  020454  042737  000200  003164            BIC     #DRY,@#REINTO+30 ;SAVED RHDS1
3778
3779  020462  005737  001100                    TST     @#$PASS         ;IS THIS FIRST PASS
3780  020466  001053                            BNE     5$              ;BRANCH IF NOT FIRST PASS
3781  020470  032737  020000  177570            BIT     #SW13,@#SWR     ;INHIBIT ERROR PRINT HIGH?
3782  020476  001047                            BNE     5$              ;BRANCH IF SW13 HIGH
3783
3784  020500  104400  020506                    TYPE    ,.+4            ;;TYPE ASCIZ STRING
3785  020504  000441                            BR      64$             ;;GET OVER THE ASCIZ
3786                                            ;;.ASCIZ      <15><12>/IF DRIVE CONNECTED "STAND BY" LAMP SHOULD BE LIT ON DRIVE
3787  020610                            64$:
3788
3789  020610  013746  001762                    MOV     @#UNIT,-(SP)    ;UNIT UNDER TEST
3790  020614  104410                            TYPOS
3791
3792                                            ;AFTER GO HAS BEEN GIVEN TO UNLOAD COMMAND
3793                                            ;SAVED REGISTERS AGAIN SO THAT COMPARISONS CAN
3794                                            ;BE DONE
3795
3796  020616  004037  041214            5$:     JSR     R0,@#SAVER      ;SAVE
```

```
3797  020622  001624                          RHWC                    ;FROM
3798  020624  002070                          WRFROM                  ;TO
3799  020626  000023                          19.                     ;NUMBER OF REGISTERS SAVED
3800
3801                                          ;AS UPPER BYTE OF RHAS CAN BE CHANGING IN A DUAL PORT
3802                                          ;OPERATION THE UPPER BYTE OF RHAS WILL BE SAVED AS IS
3803                                          ;SO THAT THE COMPARES ARE ONLY VALID FOR THE LOWER BYTE
3804  020630  113737  003161  002115          MOVB    @#REINTO+25,@#WRFROM+25;SAVE UPPER RHAS
3805
3806
3807                                          ;COMPARE REGISTERS BEFORE UNLOAD COMMAND
3808                                          ;WITH AFTER GO
3809
3810  020636  004037  041416                  JSR     R0,@#COMPAR     ;COMPARE
3811  020642  003134                          REINTO                  ;GOOD BUFFER
3812  020644  002070                          WRFROM                  ;TEST BUFFER
3813  020646  000023                          19.                     ;NUMBER
3814  020650  020656                          1$                      ;RETURN FOR ERROR
3815  020652  020656                          1$                      ;SAME
3816  020654  020676                          2$                      ;RETURN FOR GOOD COMPARISON
3817  020656  013705  045272          1$:     MOV     @#ERWORD,R5     ;GETTING READY TO INDEX
3818  020662  060505                          ADD     R5,R5           ;DOUBLE ERROR WORD
3819  020664  016537  001622  040172          MOV     RHWC-2(R5),@#REGADR ;FAILING REGISTER ADDRESS
3820
3821  020672  104001                          ERROR   1               ;IMPROPER REGISTER CHANGE
3822                                                                  ;AFTER UNLOAD COMMAND
3823                                                                  ;WITH GO IS GIVEN
3824  020674  000207                          RTS     PC              ;RETURN TO COMPARISON
3825
3826                                          ;NOW GIVE INIT AND GET ALL GO AND PIP DOWN
3827  020676  052712  000040          2$:     BIS     #CLR,@R2        ;RH INITILIZE
3828  020702  013712  001762                  MOV     @#UNIT,@R2      ;REINSTATE UNIT NUMBER
3829  020706  012777  000001  160736          MOV     #DMD,@RHMR      ;SET DIAGNOSTIC MODE BIT
3830                                                                  ;THIS ENABLES COMMANDS WITHOUT MOL
3831                                                                  ;AND HOLDS RHLA FROM MOVING
3832
3833                                          ;CHANGE REGISTERS TO EXPECTED VALUE
3834  020714  042737  000001  003142          BIC     #GO,@#REINTO+6  ;SAVED RHCS1
3835  020722  042737  020000  003164          BIC     #PIP,@#REINTO+30 ;SAVED RHDS1
3836  020730  052737  000200  003164          BIS     #DRY,@#REINTO+30 ;SAVED RHDS1
3837  020736  017737  160724  003176          MOV     @RHLA,@#REINTO+42;SAVED RHLA
3838
3839                                          ;AFTER INITIALIZE SAVE REGISTERS SO THAT
3840                                          ;COMPARES CAN BE DONE
3841
3842  020744  004037  041214                  JSR     R0,@#SAVER      ;SAVE
3843  020750  001624                          RHWC                    ;FROM
3844  020752  002070                          WRFROM                  ;TO
3845  020754  000023                          19.                     ;NUMBER OF REGISTERS SAVED
3846
3847                                          ;COMPARE REGISTERS AFTER INITIALIZE
3848                                          ;AS UPPER BYTE OF RHAS CAN BE CHANGING IN A DUAL PORT
3849                                          ;OPERATION THE UPPER BYTE OF RHAS WILL BE SAVED AS IS
3850                                          ;SO THAT THE COMPARES ARE ONLY VALID FOR THE LOWER BYTE
```

```
3851  020756  113737  003161  002115          MOVB    @#REINTO+25,@#WRFROM+25;SAVE UPPER RHAS
3852
3853
3854  020764  004037  041416                  JSR     R0,@#COMPAR     ;COMPARE
3855  020770  003134                          REINTO                  ;GOOD BUFFER
3856  020772  002070                          WRFROM                  ;TEST BUFFER
3857  020774  000023                          19.                     ;NUMBER OF REGISTERS TO BE
3858                                                                  ;COMPARED
3859  020776  021004                          3$                      ;RETURN POINT FOR ERROR
3860  021000  021004                          3$                      ;SAME
3861  021002  021024                          4$                      ;RETURN POINT FOR GOOD COMPARISON
3862
3863  021004  013705  045272          3$:     MOV     @#ERWORD,R5     ;GETTING READY TO INDEX
3864  021010  060505                          ADD     R5,R5           ;DOUBLE ERROR WORD
3865  021012  016537  001622  040172          MOV     RH%C-2(R5),@#REGADR ;FAILING REGISTER ADDRESS
3866  021020  104001                          ERROR   1               ;IMPROPER REGISTER
3867                                                                  ;CONTENTS AFTER GIVING AN
3868                                                                  ;UNLOAD COMMAND
3869  021022  000207                          RTS     PC              ;RETURN TO COMPARISON
3870  021024                          4$:                             ;GOOD COMPARISON
3871
```

```
                                        ;*****************************************************************
                                        ;*TEST 46        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
                                        ;
                                        ;*****************************************************************
021024  000004                  TST46:  SCOPE
021026  012706  001000                  MOV     #STACK,SP       ;RESET STACK
021032  012737  000046  002020          MOV     #TTNO,@#TSTNM   ;THIS SAVES TEST NUMBER
021040  004737  040522                  JSR     PC,@#CLDISK     ;INIT AND SET UP GENERAL REG.
                                                                ;AND UNIT NUMBER
021044  012777  000001  160600          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
021052  052777  000004  160572          BIS     #MINX,@RHMR     ;SET INDEX PULSE
021060  042777  000004  160564          BIC     #MINX,@RHMR     ;CLEAR INDEX

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

021066  017777  160576  160550          MOV     @RHCC,@RHCA     ;SET DESIRED CYLINDER TO RHCC
021074  013711  002054                  MOV     @#SEECOM,@R1    ;SEEK COMMAND TO RHCS1
021100  005211                          INC     @R1             ;GO TO SEEK COMMAND

                                        ;FOUR SECTOR CLOCKS ARE GIVEN TO TAKE POSITIONER OFF OFFSET POSITION
021102  012700  000004                  MOV     #4,R0           ;COUNTER
021106  012777  000011  160536  5$:     MOV     #MSTCK!DMD,@RHMR        ;SET SECTOR CLOCK
021114  012777  000001  160530          MOV     #DMD,@RHMR      ;RESET SECTOR CLOCK
021122  005300                          DEC     R0              ;COUNT
021124  001370                          BNE     5$              ;BRANCH IF NOT COMPLETE

021126  004737  040522                  JSR     PC,@#CLDISK     ;INIT AND SET UP GENERAL REG.
                                                                ;AND UNIT NUMBER
021132  012777  000001  160512          MOV     #DMD,@RHMR      ;SET DIAGNOSTIC MODE BIT
                                                                ;THIS ENABLES COMMANDS WITHOUT MOL
                                                                ;AND HOLDS RHLA FROM MOVING

021140  013777  002056  160464          MOV     @#OFSETC,@RHCS1 ;LOAD AN OFFSET BIT
021146  012777  000001  160466          MOV     #OF25,@RHOF     ;SET AN OFFSET BIT

                                        ;SAVE REGISTERS FOR COMPARISON AFTER GO
021154  004037  041214                  JSR     R0,@#SAVER      ;SAVE
021160  001624                          RHMC                    ;FROM
021162  003134                          REINTO                  ;TO
021164  000023                          19.                     ;NUMBER OF REGISTERS SAVED
```

```
                                        ;GIVE GO TO OFFSET COMMAND
021166  052777  000001  160436          BIS     #GO,@#RHCS1     ;GO TO OFFSET COMMAND

                                        ;CHANGE SAVED REGISTERS TO EXPECTED VALUES
021174  052737  000001  003142          BIS     #GO,@#REINTO+6  ;SAVED RHCS1
021202  052737  020000  003164          BIS     #PIP,@#REINTO+30 ;SAVED RHDS1
021210  042737  000200  003164          BIC     #DRY,@#REINTO+30 ;SAVED RHDS1

                                        ;AFTER GO HAS BEEN GIVEN TO OFFSET COMMAND
                                        ;SAVE REGISTERS AGAIN SO THAT COMPARISONS CAN
                                        ;BE DONE

021216  004037  041214                  JSR     R0,@#SAVER      ;SAVE
021222  001624                          RHWC                    ;FROM
021224  002070                          WRFROM                  ;TO
021226  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
021230  113737  003161  002115          MOVB    @#REINTO+25,@#WRFROM+25;SAVE UPPER RHAS

                                        ;COMPARE REGISTERS BEFORE OFFSET COMMAND
                                        ;WITH AFTER GO

021236  004037  041416                  JSR     R0,@#COMPAR     ;COMPARE
021242  003134                          REINTO                  ;GOOD BUFFER
021244  002070                          WRFROM                  ;TEST BUFFER
021246  000023                          19.                     ;NUMBER
021250  021256                          1$                      ;RETURN FOR ERROR
021252  021256                          1$                      ;SAME
021254  021276                          2$                      ;RETURN FOR GOOD COMPARISON

021256  013705  045272          1$:     MOV     @#ERWORD,R5     ;GETTING READY TO INDEX
021262  060505                          ADD     R5,R5           ;DOUBLE ERROR WORD
021264  016537  001622  040172          MOV     RHWC-2(R5),@#REGADR ;FAILING REGISTER ADDRESS
021272  104001                          ERROR   1               ;IMPROPER REGISTER CHANGE
                                                                ;AFTER OFFSET COMMAND
                                                                ;WITH GO IS GIVEN
021274  000207                          RTS     PC              ;RETURN TO COMPARISON

                                        ;NOW GIVE INIT AND GET ALL GO AND PIP DOWN

021276  052712  000040          2$:     BIS     #CLR,@R2        ;RH INITILIZE
021302  013712  001762                  MOV     @#UNIT,@R2      ;REINSTATE UNIT NUMBER
021306  012777  000001  160336          MOV     #DMD,@RHMR      ;SET DIAGNOSTIC MODE BIT
                                                                ;THIS ENABLES COMMANDS WITHOUT MOL
                                                                ;AND HOLDS RHLA FROM MOVING

                                        ;CHANGE REGISTERS TO EXPECTED VALUE
021314  042737  000001  003142          BIC     #GO,@#REINTO+6  ;SAVED RHCS1
021322  042737  000001  003152          BIC     #0F25,@#REINTO+16;SAVED RHOF
021330  042737  020000  003164          BIC     #PIP,@#REINTO+30 ;SAVED RHDS1
```

```
3980  021336  052737  000200  003164          BIS     #DRY,@#REINTO+30 ;SAVED RHDS1
3981  021344  017737  160316  003176          MOV     @RHLA,@#REINTO+42;SAVED RHLA
3982
3983                                          ;AFTER INITIALIZE SAVE REGISTERS SO THAT
3984                                          ;COMPARES CAN BE DONE
3985
3986  021352  004037  041214                  JSR     R0,@#SAVER      ;SAVE
3987  021356  001624                          RHWC                    ;FROM
3988  021360  002070                          WRFROM                  ;TO
3989  021362  000023                          19.                     ;NUMBER OF REGISTERS SAVED
3990
3991                                          ;AS UPPER BYTE OF RHAS CAN BE CHANGING IN A DUAL PORT
3992                                          ;OPERATION THE UPPER BYTE OF RHAS WILL BE SAVED AS IS
3993                                          ;SO THAT THE COMPARES ARE ONLY VALID FOR THE LOWER BYTE
3994  021364  113737  003161  002115          MOVB    @#REINTO+25,@#WRFROM+25;SAVE UPPER RHAS
3995
3996                                          ;COMPARE REGISTERS AFTER INITIALIZE
3997  021372  004037  041416                  JSR     R0,@#COMPAR     ;COMPARE
3998  021376  003134                          REINTO                  ;GOOD BUFFER
3999  021400  002070                          WRFROM                  ;TEST BUFFER
4000  021402  000023                          19.                     ;NUMBER OF REGISTERS TO BE
4001                                                                  ;COMPARED
4002  021404  021412                          3$                      ;RETURN POINT FOR ERROR
4003  021406  021412                          3$                      ;SAME
4004  021410  021432                          4$                      ;RETURN POINT FOR GOOD COMPARISON
4005
4006  021412  013705  045272          3$:     MOV     @#ERWORD,R5     ;GETTING READY TO INDEX
4007  021416  060505                          ADD     R5,R5           ;DOUBLE ERROR WORD
4008  021420  016537  001622  040172          MOV     RHWC-2(R5),@#REGADR ;FAILING REGISTER ADDRESS
4009  021426  104001                          ERROR   1               ;IMPROPER REGISTER
4010                                                                  ;CONTENTS AFTER GIVING AN
4011                                                                  ;INITIALIZE FOLLOWING A
4012                                                                  ;OFFSET COMMAND
4013  021430  000207                          RTS     PC              ;RETURN TO COMPARISON
4014
4015  021432                          4$:                             ;GOOD COMPARISON
```

```
                                        ;*****************************************************************
                                        ;*TEST 47       RETURN TO CENTER LINE COMMAND TEST

                                        ;*      THE RETURN TO CENTER LINE 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

                                        ;*****************************************************************
021432  000004                  TST47:  SCOPE
021434  012706  001000                  MOV     #STACK,SP       ;RESET STACK
021440  012737  000047  002020          MOV     #TTNO,@#TSTNM   ;THIS SAVES TEST NUMBER
021446  004737  040522                  JSR     PC,@#CLDISK     ;INIT AND SET UP GENERAL REG.
                                                                ;AND UNIT NUMBER
021452  012777  000001  160172          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
021460  052777  000004  160164          BIS     #MINX,@RHMR     ;SET INDEX PULSE
021466  042777  000004  160156          BIC     #MINX,@RHMR     ;CLEAR INDEX

021474  013777  002060  160130          MOV     @#RETCL,@RHCS1  ;LOAD RETURN TO CENTER LINE COMMAND INTO RHCS1

                                        ;SAVE REGISTERS FOR COMPARISON AFTER GO
021502  004037  041214                  JSR     R0,@#SAVER      ;SAVE
021506  001624                          RHMC                    ;FROM
021510  003134                          REINTO                  ;TO
021512  000023                          19.                     ;NUMBER OF REGISTERS SAVED

                                        ;GIVE GO TO RETURN TO CENTER LINE COMMAND
021514  052777  000001  160110          BIS     #GO,@RHCS1      ;GO TO RETURN TO CENTER COMMAND


                                        ;FOUR SECTOR CLOCKS ARE GIVEN TO TAKE POSITIONER TO CENTER LINE
021522  012700  000004                  MOV     #4,R0           ;COUNTER
021526  012777  000011  160116  5$:     MOV     #MSTCK!DMD,@RHMR;SET SECTOR CLOCK
021534  012777  000001  160110          MOV     #DMD,@RHMR      ;RESET SECTOR CLOCK
021542  005300                          DEC     R0              ;COUNT
021544  001370                          BNE     5$              ;BRANCH IF NOT COMPLETE


                                        ;CHANGE SAVED REGISTERS TO EXPECTED VALUES
021546  052737  000001  003142          BIS     #GO,@#REINTO+6  ;SAVED RHCS1
021554  052737  020000  003164          BIS     #PIP,@#REINTO+30 ;SAVED RHDS1
021562  042737  000200  003164          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
021570  004037  041214                  JSR     R0,@#SAVER      ;SAVE
```

```
4070  021574  001624                          RHWC                    ;FROM
4071  021576  002070                          WRFROM                  ;TO
4072  021600  000023                          19.                     ;NUMBER OF REGISTERS SAVED
4073
4074                                          ;AS UPPER BYTE OF RHAS CAN BE CHANGING IN A DUAL PORT
4075                                          ;OPERATION THE UPPER BYTE OF RHAS WILL BE SAVED AS IS
4076                                          ;SO THAT THE COMPARES ARE ONLY VALID FOR THE LOWER BYTE
4077  021602  113737  003161  002115          MOVB    @#REINTO+25,@#WRFROM+25;SAVE UPPER RHAS
4078
4079
4080                                          ;COMPARE REGISTERS BEFORE RETURN TO CENTER LINE COMMAND
4081                                          ;WITH AFTER GO
4082  021610  004037  041416                  JSR     R0,@#COMPAR     ;COMPARE
4083  021614  003134                          REINTO                  ;GOOD BUFFER
4084  021616  002070                          WRFROM                  ;TEST BUFFER
4085  021620  000023                          19.                     ;NUMBER
4086  021622  021630                          1$                      ;RETURN FOR ERROR
4087  021624  021630                          1$                      ;SAME
4088  021626  021650                          2$                      ;RETURN FOR GOOD COMPARISON
4089
4090  021630  013705  045272          1$:     MOV     @#ERWORD,R5     ;GETTING READY TO INDEX
4091  021634  060505                          ADD     R5,R5           ;DOUBLE ERROR WORD
4092  021636  016537  001622  040172          MOV     RHWC-2(R5),@#REGADR ;FAILING REGISTER ADDRESS
4093  021644  104001                          ERROR   1               ;IMPROPER REGISTER CHANGE
4094                                                                  ;AFTER RETURN TO CENTER LINE COMMAND
4095                                                                  ;WITH GO IS GIVEN
4096  021646  000207                          RTS     PC              ;RETURN TO COMPARISON
4097
4098                                          ;NOW GIVE INIT AND GET ALL GO AND PIP DOWN
4099
4100  021650  052712  000040          2$:     BIS     #CLR,@R2        ;RH INITILIZE
4101  021654  013712  001762                  MOV     @#UNIT,@R2      ;REINSTATE UNIT NUMBER
4102  021660  012777  000001  157764          MOV     #DMD,@RHMR      ;SET DIAGNOSTIC MODE BIT
4103                                                                  ;THIS ENABLES COMMANDS WITHOUT MOL
4104                                                                  ;AND HOLDS RHLA FROM MOVING
4105
4106                                          ;CHANGE REGISTERS TO EXPECTED VALUE
4107  021666  042737  000001  003142          BIC     #GO,@#REINTO+6  ;SAVED RHCS1
4108  021674  042737  020000  003164          BIC     #PIP,@#REINTO+30 ;SAVED RHDS1
4109  021702  052737  000200  003164          BIS     #DRY,@#REINTO+30 ;SAVED RHDS1
4110  021710  017737  157752  003176          MOV     @RHLA,@#REINTO+42;SAVED RHLA
4111
4112                                          ;AFTER INITIALIZE SAVE REGISTERS SO THAT
4113                                          ;COMPARES CAN BE DONE
4114  021716  004037  041214                  JSR     R0,@#SAVER      ;SAVE
4115  021722  001624                          RHWC                    ;FROM
4116  021724  002070                          WRFROM                  ;TO
4117  021726  000023                          19.                     ;NUMBER OF REGISTERS SAVED
4118
4119                                          ;AS UPPER BYTE OF RHAS CAN BE CHANGING IN A DUAL PORT
4120                                          ;OPERATION THE UPPER BYTE OF RHAS WILL BE SAVED AS IS
4121                                          ;SO THAT THE COMPARES ARE ONLY VALID FOR THE LOWER BYTE
4122  021730  113737  003161  002115          MOVB    @#REINTO+25,@#WRFROM+25;SAVE UPPER RHAS
4123
```

```
                                        ;COMPARE REGISTERS AFTER INITIALIZE
021736  004037  041416                  JSR     R0,@#COMPAR     ;COMPARE
021742  003134                          REINTO                  ;GOOD BUFFER
021744  002070                          WRFROM                  ;TEST BUFFER
021746  000023                          19.                     ;NUMBER OF REGISTERS TO BE
                                                                ;COMPARED
021750  021756                          3$                      ;RETURN POINT FOR ERROR
021752  021756                          3$                      ;SAME
021754  021776                          4$                      ;RETURN POINT FOR GOOD COMPARISON

021756  013705  045272          3$:     MOV     @#ERWORD,R5     ;GETTING READY TO INDEX
021762  060505                          ADD     R5,R5           ;DOUBLE ERROR WORD
021764  016537  001622  040172          MOV     RHWC-2(R5),@#REGADR ;FAILING REGISTER ADDRESS
021772  104001                          ERROR   1               ;IMPROPER REGISTER
                                                                ;CONTENTS AFTER GIVING AN
                                                                ;INITIALIZE FOLLOWING A RETURN TO
                                                                ;CENTER LINE COMMAND
021774  000207                          RTS     PC              ;RETURN TO COMPARISON

021776                          4$:                             ;GOOD COMPARISON
```

```
4147
4148
4149                                    ;*****************************************************************
4150                                    ;*TEST 50       RECALIBRATE COMMAND TEST
4151
4152                                    ;*      THE RECALIBRATE COMMAND WILL BE LOADED INTO RHCS1 WITH GO
4153                                    ;*      THEN ALL REGISTERS WILL BE CHECKED
4154                                    ;*      RH CLEAR WILL BE GIVEN
4155                                    ;*      THEN ALL REGISTERS WILL BE CHECKED
4156
4157                                    ;*****************************************************************
4158  021776  000004                    TST50:  SCOPE
4159  022000  012706  001000                    MOV     #STACK,SP       ;RESET STACK
4160  022004  012737  000050  002020            MOV     #TTNO,@#TSTNM   ;THIS SAVES TEST NUMBER
4161  022012  004737  040522                    JSR     PC,@#CLDISK     ;INIT AND SET UP GENERAL REG.
4162                                                                    ;AND UNIT NUMBER
4163  022016  012777  000001  157626            MOV     #DMD,@RHMR      ;SET DIAGNOSTIC MODE BIT
4164                                                                    ;THIS ENABLES COMMANDS WITHOUT MOL
4165                                                                    ;AND HOLDS RHLA FROM MOVING
4166
4167
4168                                    ;GIVE ONE INDEX PULSE TO CLEAR RHLA BEFORE THE START OF THIS TEST
4169
4170  022024  052777  000004  157620            BIS     #MINX,@RHMR     ;SET INDEX PULSE
4171  022032  042777  000004  157612            BIC     #MINX,@RHMR     ;CLEAR INDEX
4172
4173  022040  013777  002030  157564            MOV     @#RECALI,@RHCS1 ;LOAD RECALIBRATE COMMAND INTO RHCS1
4174
4175                                    ;SAVE REGISTERS FOR COMPARISON AFTER GO
4176  022046  004037  041214                    JSR     R0,@#SAVER      ;SAVE
4177  022052  001624                            RHMC                    ;FROM
4178  022054  003134                            REINTO                  ;TO
4179  022056  000023                            19.                     ;NUMBER OF REGISTERS SAVED
4180
4181                                    ;GIVE GO TO RECALIBRATE COMMAND
4182  022060  052777  000001  157544            BIS     #GO,@RHCS1      ;GO TO RECALIBRATE COMMAND
4183
4184
4185                                    ;FOUR SECTOR CLOCKS ARE GIVEN TO TAKE POSITIONER TO CYLINDER 0
4186  022066  012700  000004                    MOV     #4,R0           ;COUNTER
4187  022072  012777  000011  157552  5$:       MOV     #MSTCK!DMD,@RHMR;SET SECTOR CLOCK
4188  022100  012777  000001  157544            MOV     #DMD,@RHMR      ;RESET SECTOR CLOCK
4189  022106  005300                            DEC     R0              ;COUNT
4190  022110  001370                            BNE     5$              ;BRANCH IF NOT COMPLETE
4191
4192
4193                                    ;CHANGE SAVED REGISTERS TO EXPECTED VALUES
4194  022112  052737  000001  003142            BIS     #GO,@#REINTO+6  ;SAVED RHCS1
4195  022120  052737  020000  003164            BIS     #PIP,@#REINTO+30 ;SAVED RHDS1
4196  022126  042737  000200  003164            BIC     #DRY,@#REINTO+30 ;SAVED RHDS1
4197
4198                                    ;AFTER GO HAS BEEN GIVEN TO RECALIBRATE COMMAND
4199                                    ;SAVE REGISTERS AGAIN SO THAT COMPARISONS CAN
4200                                    ;BE DONE
```

```
4201  022134  004037  041214              JSR     R0,@#SAVER      ;SAVE
4202  022140  001624                      RHWC                    ;FROM
4203  022142  002070                      WRFROM                  ;TO
4204  022144  000023                      19.                     ;NUMBER OF REGISTERS SAVED
4205
4206                                      ;AS UPPER BYTE OF RHAS CAN BE CHANGING IN A DUAL PORT
4207                                      ;OPERATION THE UPPER BYTE OF RHAS WILL BE SAVED AS IS
4208                                      ;SO THAT THE COMPARES ARE ONLY VALID FOR THE LOWER BYTE
4209  022146  113737  003161  002115      MOVB    @#REINT0+25,@#WRFROM+25;SAVE UPPER RHAS
4210
4211
4212                                      ;COMPARE REGISTERS BEFORE RECALIBRATE COMMAND
4213                                      ;WITH AFTER GO
4214  022154  004037  041416              JSR     R0,@#COMPAR     ;COMPARE
4215  022160  003134                      REINT0                  ;GOOD BUFFER
4216  022162  002070                      WRFROM                  ;TEST BUFFER
4217  022164  000023                      19.                     ;NUMBER
4218  022166  022174                      1$                      ;RETURN FOR ERROR
4219  022170  022174                      1$                      ;SAME
4220  022172  022214                      2$                      ;RETURN FOR GOOD COMPARISON
4221
4222  022174  013705  045272      1$:     MOV     @#ERWORD,R5     ;GETTING READY TO INDEX
4223  022200  060505                      ADD     R5,R5           ;DOUBLE ERROR WORD
4224  022202  016537  001622  040172      MOV     RHWC-2(R5),@#REGADR ;FAILING REGISTER ADDRESS
4225  022210  104001                      ERROR   1               ;IMPROPER REGISTER CHANGE
4226                                                              ;AFTER RECALIBRATE COMMAND
4227                                                              ;WITH GO IS GIVEN
4228  022212  000207                      RTS     PC              ;RETURN TO COMPARISON
4229
4230                                      ;NOW GIVE INIT AND GET ALL GO AND PIP DOWN
4231
4232  022214  052712  000040      2$:     BIS     #CLR,@R2        ;RH INITILIZE
4233  022220  013712  001762              MOV     @#UNIT,@R2      ;REINSTATE UNIT NUMBER
4234  022224  012777  000001  157420      MOV     #DMD,@RHMR      ;SET DIAGNOSTIC MODE BIT
4235                                                              ;THIS ENABLES COMMANDS WITHOUT MOL
4236                                                              ;AND HOLDS RHLA FROM MOVING
4237
4238                                      ;CHANGE REGISTERS TO EXPECTED VALUE
4239  022232  042737  000001  003142      BIC     #GO,@#REINT0+6  ;SAVED RHCS1
4240  022240  042737  020000  003164      BIC     #PIP,@#REINT0+30 ;SAVED RHDS1
4241  022246  052737  000200  003164      BIS     #DRY,@#REINT0+30 ;SAVED RHDS1
4242  022254  017737  157406  003176      MOV     @RHLA,@#REINT0+42;SAVED RHLA
4243
4244                                      ;AFTER INITIALIZE SAVE REGISTERS SO THAT
4245                                      ;COMPARES CAN BE DONE
4246  022262  004037  041214              JSR     R0,@#SAVER      ;SAVE
4247  022266  001624                      RHWC                    ;FROM
4248  022270  002070                      WRFROM                  ;TO
4249  022272  000023                      19.                     ;NUMBER OF REGISTERS SAVED
4250
4251                                      ;AS UPPER BYTE OF RHAS CAN BE CHANGING IN A DUAL PORT
4252                                      ;OPERATION THE UPPER BYTE OF RHAS WILL BE SAVED AS IS
4253                                      ;SO THAT THE COMPARES ARE ONLY VALID FOR THE LOWER BYTE
4254  022274  113737  003161  002115      MOVB    @#REINT0+25,@#WRFROM+25;SAVE UPPER RHAS
```

N11


```
                                                        ;COMPARE REGISTERS AFTER INITIALIZE
022302  004037  041416                                  JSR     R0,@#COMPAR     ;COMPARE
022306  003134                                          REINTO                  ;GOOD BUFFER
022310  002070                                          WRFROM                  ;TEST BUFFER
022312  000023                                          19.                     ;NUMBER OF REGISTERS TO BE
                                                                                ;COMPARED
022314  022322                                          3$                      ;RETURN POINT FOR ERROR
022316  022322                                          3$                      ;SAME
022320  022342                                          4$                      ;RETURN POINT FOR GOOD COMPARISON

022322  013705  045272          3$:     MOV     @#ERWORD,R5     ;GETTING READY TO INDEX
022326  060505                          ADD     R5,R5           ;DOUBLE ERROR WORD
022330  016537  001622  040172         MOV     RH#MC-2(R5),@#REGADR ;FAILING REGISTER ADDRESS
022336  104001                          ERROR   1               ;IMPROPER REGISTER
                                                                ;CONTENTS AFTER GIVING AN
                                                                ;INITIALIZE FOLLOWING A
                                                                ;RECALIBRATE COMMAND
022340  000207                          RTS     PC              ;RETURN TO COMPARISON

022342                          4$:                             ;GOOD COMPARISON
```

```
4277
4278
4279                                    ;*****************************************************************
4280                                    ;*TEST 51        RELEASE COMMAND TEST
4281
4282                                    ;*      THE RELEASE COMMAND WILL BE LOADED INTO RHCS1 WITH GO
4283                                    ;*      THEN ALL REGISTERS WILL BE CHECKED
4284                                    ;*      RH CLEAR WILL BE GIVEN
4285                                    ;*      THEN ALL REGISTERS WILL BE CHECKED
4286
4287                                    ;*****************************************************************
4288  022342  000004                   TST51:  SCOPE
4289  022344  012706  001000                   MOV     #STACK,SP       ;RESET STACK
4290  022350  012737  000051  002020           MOV     #TTNO,@#TSTNM   ;THIS SAVES TEST NUMBER
4291  022356  004737  040522                   JSR     PC,@#CLDISK     ;INIT AND SET UP GENERAL REG.
4292                                                                   ;AND UNIT NUMBER
4293  022362  012777  000001  157262           MOV     #DMD,@RHMR      ;SET DIAGNOSTIC MODE BIT
4294                                                                   ;THIS ENABLES COMMANDS WITHOUT MOL
4295                                                                   ;AND HOLDS RHLA FROM MOVING
4296
4297  022370  013777  002034  157234           MOV     @#RELEASE,@RHCS1 ;LOAD RELEASE COMMAND INTO RHCS1
4298
4299                                           ;SAVE REGISTERS FOR COMPARISON AFTER GO
4300  022376  004037  041214                   JSR     R0,@#SAVER      ;SAVE
4301  022402  001624                           RHMC                    ;FROM
4302  022404  003134                           REINTO                  ;TO
4303  022406  000023                           19.                     ;NUMBER OF REGISTERS SAVED
4304
4305                                           ;GIVE GO TO RELEASE COMMAND
4306  022410  052777  000001  157214           BIS     #GO,@RHCS1      ;GO TO RELEASE COMMAND
4307  022416  052777  000001  157226           BIS     #DMD,@RHMR      ;SET DMD TO HOLD RHLA
4308
4309                                           ;CHANGE SAVED REGISTERS TO EXPECTED VALUES
4310                                           ;AFTER GO HAS BEEN GIVEN TO RELEASE COMMAND
4311  022424  052737  000001  003162           BIS     #DMD,@#REINTO+26;SAVED RHMR
4312  022432  017737  157230  003176           MOV     @RHLA,@#REINTO+42;SAVED RHLA
4313
4314
4315                                           ;SAVE REGISTERS AGAIN SO THAT COMPARISONS CAN
4316                                           ;BE DONE
4317  022440  004037  041214                   JSR     R0,@#SAVER      ;SAVE
4318  022444  001624                           RHMC                    ;FROM
4319  022446  002070                           WRFROM                  ;TO
4320  022450  000023                           19.                     ;NUMBER OF REGISTERS SAVED
4321
4322                                           ;AS UPPER BYTE OF RHAS CAN BE CHANGING IN A DUAL PORT
4323                                           ;OPERATION THE UPPER BYTE OF RHAS WILL BE SAVED AS IS
4324                                           ;SO THAT THE COMPARES ARE ONLY VALID FOR THE LOWER BYTE
4325  022452  113737  003161  002115           MOVB    @#REINTO+25,@#WRFROM+25;SAVE UPPER RHAS
4326
4327
4328                                           ;COMPARE REGISTERS BEFORE RELEASE COMMAND
4329                                           ;WITH AFTER GO
4330  022460  004037  041416                   JSR     R0,@#COMPAR     ;COMPARE
```

```
4331   022464  003134                          REINTO                  ;GOOD BUFFER
4332   022466  002070                          WRFROM                  ;TEST BUFFER
4333   022470  000023                          19.                     ;NUMBER
4334   022472  022500                          1$                      ;RETURN FOR ERROR
4335   022474  022500                          1$                      ;SAME
4336   022476  022520                          2$                      ;RETURN FOR GOOD COMPARISON
4337
4338   022500  013705  045272          1$:     MOV     @#ERWORD,R5     ;GETTING REHYD TO INDEX
4339   022504  060505                          ADD     R5,R5           ;DOUBLE ERROR WORD
4340   022506  016537  001622  040172          MOV     RHWC-2(R5),@#REGADR ;FAILING REGISTER ADDRESS
4341   022514  104001                          ERROR   1               ;IMPROPER REGISTER CHANGE
4342                                                                   ;AFTER RELEASE COMMAND
4343                                                                   ;WITH GO IS GIVEN
4344   022516  000207                          RTS     PC              ;RETURN TO COMPARISON
4345   022520                          2$:
4346
```

```
                                        ;*****************************************************************
                                        ;*TEST 52       MAKE CURRENT CYLINDER = 0
                                        ;*****************************************************************
022520  000004                        TST52:  SCOPE
022522  012706  001000                        MOV     #STACK,SP       ;RESET STACK
022526  004737  040522                        JSR PC,@#CLDISK         ;INIT DRIVE
022532  012777  000001  157112                MOV     #DMD,@RHMR      ;SET DIAGNOSTIC MODE
022540  004037  043074                        JSR     R0,@#MAKECYL    ;SUBROUTINE TO GIVE A SEEK
                                                                        ;COMMAND FOLOWED BY A INIT
                                                                        ;THIS SHUOLD CHANGE RHCC
022544  000000                                0                       ;CHANGE RHCC TO 0


                                        ;*****************************************************************
                                        ;*TEST 53       LOOK AHEAD REGISTER

                                        ;*      A SEARCH COMMAND IS GIVEN FOR CYLINDER 0, TRACK 0, SECTOR 21.
                                        ;*      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 AFTER 256 MORE BYTES
                                        ;*      THE SECTOR COUNT FIELD IS CHECKED AFTER EACH SECTOR
                                        ;*      AT THE END ALL REGISTERS ARE CHECKED

                                        ;*****************************************************************
022546  000004                        TST53:  SCOPE
022550  012706  001000                        MOV     #STACK,SP       ;RESET STACK
022554  012737  000053  002020                MOV     #TTNO,@#TSTNM   ;THIS SAVES TEST NUMBER
022562  004737  040522                        JSR     PC,@#CLDISK     ;INIT AND SET UP GENERAL REGISTERS

                                        ;THESE ARE REGULAR SET UPS FOR SEARCH COMMAND
022566  012777  000025  157042                MOV     #21.,@RHDST     ;DESIRED SECTOR/TRACK REGISTER
                                                                        ;TRACK 0 SECTOR 21
022574  005077  157044                        CLR     @RHCA           ;DESIRED CYLINDER =0
022600  012777  010000  157034                MOV     #FMT22,@RHOF    ;FORMAT BIT=1 (16 BITS PER WORD)
022606  013711  002036                        MOV     @#SERCH,@R1     ;FILL SEARCH COMMAND IN RHCS1

                                        ;NOW SAVE REGISTERS STARTING FROM RHMC IN WRITE FROM BUFFER
022612  004037  041214                        JSR     R0,@#SAVER      ;SAVE REGISTERS FOR COMPARISON
                                                                        ;AT THE END OF THE SEARCH
022616  001624                                RHMC                    ;START SAVING FROM RHMC
022620  003134                                REINTO                  ;SAVE INTO REINTO
022622  000023                                19.                     ;NUMBER OF REGISTERS SAVED

022624  004737  040556                        JSR     PC,@#CHECKT     ;CHECK THAT DVA,RDY,DPR,DRY = 1
022630  104400  057176                        TYPE    ,CPHALT         ;AND THAT NO OTHERS = 1.  CANNOT CON-
                                                                        ;TINUE TESTING IF BOTH AREN'T TRUE
022634  000000                                HALT                    ;STOP THE TEST

                                        ;NOW THE DIAGNOSTIC MODE BIT WILL BE SET
                                        ;AND THE SEARCH OPERATION STARTED
```

```
4401
4402  022636  005037  001172                  CLR     @#STMP1         ;THIS WILL HAVE THE EXPECTED
4403                                                                    ;VALUE OF RHLA REGISTER
4404
4405  022642  013700  001652                  MOV     @#RHMR,R0       ;NOW R0 HAS MAINTENANCE REG. ADDR.
4406  022646  017703  156764                  MOV     @RHDST,R3       ;GET DESIRED SECTOR/TRACK REG.
4407  022652  042703  177400                  BIC     #177400,R3      ;GET SECTOR ONLY
4408  022656  010337  051202                  MOV     R3,@#SECTR      ;DUPLICATE SECTOR
4409  022662  012710  000001                  MOV     #DMD,@R0        ;S
4410  022666  052777  000001  156736          BIS     #GO,@RHCS1      ;GO
4411  022674  052710  000010                  BIS     #MSTCK,@R0      ;SET SECTOR CLOCK
4412  022700  042710  000010                  BIC     #MSTCK,@R0      ;CLEAR SECTOR CLOCK
4413  022704  000240                          NOP                     ;ALLOW TIME BETWEEN SECTOR CLOCKS
4414  022706  052710  000010                  BIS     #MSTCK,@R0      ;SET SECTOR CLOCK
4415  022712  042710  000010                  BIC     #MSTCK,@R0      ;CLEAR SECTOR CLOCK
4416  022716  000240                          NOP                     ;ALLOW TIME BETWEEN SECTOR CLOCKS
4417  022720  052710  000014                  BIS     #MINX!MSTCK,@R0 ;SET INDEX AND SECTOR CLOCK
4418  022724  012710  000001                  MOV     #DMD,@R0        ;RESET INDEX AND SECTOR CLOCK
4419  022730  005703                          TST     R3              ;IF SECTOR REQUIRED JUMP OUT
4420  022732  001555                          BEQ     11$             ;BRANCH OF SECTOR ZERO REQUIRED
4421
4422                                          ;AFTER THE INDEX PULSE RHLA WILL BE CHECKED TO BE ZERO
4423                                          ;AND STMP4 WILL BE SET UP TO COUNT BYTES
4424  022734  012737  001140  001200  1$:     MOV     #608.,@#STMP4   ;THERE ARE 608 BYTES PER SECTOR
4425  022742  017737  156720  001126          MOV     @RHLA,@#SBODAT  ;SAVE RHLA
4426  022750  017737  156662  001706          MOV     @RHDST,@#DST    ;SAVE DESIRED SECTOR TRACK
4427  022756  023737  001172  001126          CMP     @#STMP1,@#SBODAT ;RHLA SHOULD BE HAVE EXTENSION
4428                                                                  ;FIELD EQUAL TO ZERO
4429  022764  001414                          BEQ     2$              ;BRANCH IF GOOD
4430  022766  013737  051202  001174          MOV     @#SECTR,@#STMP2 ;GET SECTOR SOUGHT
4431  022774  160337  001174                  SUB     R3,@#STMP2      ;STMP2 NOW HAS PRESENT SECTOR
4432  023000  012746  001140                  MOV     #608.,-(SP)     ;NUMBER OF BYTES PER SECTOR
4433  023004  163716  001200                  SUB     @#STMP4,(SP)    ;(SP)HAS PRESENT BYTE NUMBER
4434  023010  012637  001176                  MOV     (SP)+,@#STMP3   ;PRESENT BYTE NUMBER
4435  023014  104024                          ERROR   24              ;LOOK AHEAD REGISTER AT THE BEGINING OF A
4436                                                                  ;SECTOR IS IN ERROR
4437
4438
4439                                          ;NOW THE 304 WORDS WILL START
4440                                          ;FOR FIRST BYTE CLOCK WILL BE INDEPENDENT OF
4441                                          ;SECTOR CLOCK THEN IT WILL COINCIDE FOREVER TILL
4442                                          ;THE BEGINNING OF NEXT SECTOR
4443
4444                                          ;ONE WORD ONLY THAT IS TWO BYTES
4445
4446  023016  012702  000010          2$:     MOV     #8.,R2          ;BYTE
4447  023022  012705  000002                  MOV     #2,R5           ;BYTES PER WORD
4448  023026  000404                          BR      4$
4449  023030  052710  000012          3$:     BIS     #MSTCK!MCLK,@R0 ;SET SECTOR AND CLOCK
4450  023034  042710  000012                  BIC     #MSTCK!MCLK,@R0 ;CLEAR SECTOR AND CLOCK
4451  023040  052710  000002          4$:     BIS     #MCLK,@R0       ;SET CLOCK
4452  023044  042710  000002                  BIC     #MCLK,@R0       ;CLEAR CLOCK
4453  023050  005302                          DEC     R2              ;BYTE COUNTER
4454  023052  001372                          BNE     4$              ;BRANCH IF BYTE NOT COMPLETE
```

```
4455  023054  005337  001200                DEC     @#STMP4         ;BYTE COUNT DOWN
4456  023060  012702  000007                MOV     #7,R2           ;SETUP FOR SECOND BYTE
4457  023064  005305                        DEC     R5              ;IS WORD COMPLETE?
4458  023066  001360                        BNE     3$              ;BRENCH IF NOT COMPLETE
4459                                                                ;TO GIVE SECTOR CLOCK AND CLOCK
4460
4461                                        ;NOW 303 WORDS ARE LEFT ALL ARE IDENTICAL
4462                                        ;THAT IS 606 IDENTICAL BYTES WILL BE GIVEN
4463                                        ;RHLA WILL BE CHECKED STAR TO COUNT AFTER
4464                                        ;BEGINNING OF SECTOR PULSE
4465                                        ;AFTER 128 BYTES (2 BYTES ARE ALREADY GIVEN)
4466                                        ;SO 127 MORE
4467                                        ;THEN RHLA WILL BE CHECKED AFTER 128 MORE BYTES
4468                                        ;THEN RHLA WILL BE CHECKED AFTER 256 MORE BYTES
4469                                        ;THEN THE TOTAL OF 508 BYTES WILL BE COMPLETED
4470                                        ;AND RHLA WILL BE MADE READY FOR NEXT SECTOR
4471                                        ;AND RHLA WILL BE CHECKED
4472
4473  023070  012705  000100                MOV     #64.,R5         ;R5 WILL KEEP TRACK WHEN
4474                                                                ;EXTENSION FIELD IS TO BE CHECKED
4475  023074  012701  000177                MOV     #127.,R1        ;FIRST TIME CHECK EXTENSION FIELD
4476                                                                ;AFTER 127 MORE BYTES
4477  023100  012702  000007        5$:     MOV     #7,R2           ;CLOCKS PER BYTE COUNTER
4478  023104  052710  000012                BIS     #MSTCK!MCLK,@R0 ;SET SECTOR CLOCK AND CLOCK
4479  023110  042710  000012                BIC     #MSTCK!MCLK,@R0 ;CLEAR SECTOR CLOCK AND CLOCK
4480  023114  052710  000002        6$:     BIS     #MCLK,@R0       ;SET CLOCK
4481  023120  042710  000002                BIC     #MCLK,@R0       ;RESET CLOCK
4482  023124  005302                        DEC     R2              ;COUNT DOWN CLOCKS PER BYTE
4483  023126  001372                        BNE     6$              ;BRANCH IF BYTE NOT COMPLETE
4484  023130  005337  001200                DEC     @#STMP4         ;COUNT DOWN BYTES
4485  023134  001436                        BEQ     10$             ;BRANCHOUT IF 608 BYTES DONE
4486  023136  005301                        DEC     R1              ;COUNT DOWN NUMBER OF BYTES
4487                                                                ;TO CHECK EXTENSION FIELD
4488  023140  001357                        BNE     5$              ;BRANCH IF EXTENSION FIELD NOT
4489                                                                ;TO BE CHECKED YET
4490
4491                                        ;NOW THE EXTENSION FIELD OF THE LOOK AHEAD REGISTER
4492                                        ;WILL BE CHECKED
4493
4494  023142  062737  000020  001172        ADD     #20,@#STMP1     ;GET TO THE NEXT EXTENSION
4495  023150  017737  156512  001126        MOV     @RHLA,@#SBDDAT  ;GET RHLA FOR COMPARISON
4496
4497  023156  017737  156454  001706        MOV     @RHDST,@#DST    ;SAVE DESIRED SECTOR TRACK
4498  023164  023737  001172  001126        CMP     @#STMP1,@#SBDDAT ;CHECK VALUE OF RHLA
4499  023172  001414                        BEQ     7$              ;BRANCH IF GOOD
4500  023174  013737  051202  001174        MOV     @#SECTR,@#STMP2 ;GET SECTOR SOUGHT
4501  023202  160337  001174                SUB     R3,@#STMP2      ;STMP2 NOW HAS PRESENT SECTOR
4502  023206  012746  001140                MOV     #608.,-(SP)     ;NUMBER OF BYTES PER SECTOR
4503  023212  163716  001200                SUB     @#STMP4,(SP)    ;(SP) HAS PRESENT BYTE NUMBER
4504  023216  012637  001176                MOV     (SP)+,@#STMP3   ;PRESENT BYTE NUMBER
4505  023222  104025                        ERROR   25              ;LOOK AHEAD ERROR IN THE MIDDLE
4506                                                                ;OF A SECTOR IS IN ERROR
4507
4508  023224  060505                7$:     ADD     R5,R5           ;GET NEXT STEP TO CHECK EXTENSION FIE_D
```

```
4509  023226  010501                          MOV     R5,R1           ;PUT IN COUNTER
4510  023230  000723                          BR      5$              ;BRANCH BACK SECTOR
4511                                                                  ;IS NOT COMPLETE
4512  023232  062737  000020  001172  10$:    ADD     #20,@#$TMP1
4513  023240  052710  000010                  BIS     #MSTCK,@R0      ;THESE TWO INSTRUCTIONS GIVE
4514  023244  042710  000010                  BIC     #MSTCK,@R0      ;ONE SECTOR CLOCK EXTRA
4515  023250  000240                          NOP                     ;ALLOW TIME BETWEEN SECTOR CLOCK
4516  023252  052710  000010                  BIS     #MSTCK,@R0      ;THESE TWO INSTRUCTIONS GIVE
4517  023256  042710  000010                  BIC     #MSTCK,@R0      ;ONE SECTOR CLOCK EARLY
4518                                                                  ;BEFORE THE NEXT SECTOR
4519  023262  005303                          DEC     R3              ;IS REQUIRED NO OF SECTORS COMPLETE
4520  023264  001223                          BNE     1$              ;BRANCH IF NOT
4521
4522                                          ;NOW THE REQUIRED SECTOR IS REACHED
4523                                          ;ONE SECTOR CLOCK WILL BE GIVEN TO GET SECTOR PULSE
4524                                          ;DOWN AND HENCE ATA UP
4525
4526  023266  012702  000010          11$:    MOV     #8.,R2          ;8 CLOCKS
4527  023272  052710  000002          12$:    BIS     #MCLK,@R0       ;SET CLOCK
4528  023276  042710  000002                  BIC     #MCLK,@R0       ;CLEAR CLOCKS
4529  023302  005302                          DEC     R2              ;COUND DOWN
4530  023304  001372                          BNE     12$             ;BRANCH IF 8 NOT DONE
4531  023306  052710  000012                  BIS     #MSTCK!MCLK,@R0 ;SET SECTOR AND CLOCK
4532  023312  042710  000012                  BIC     #MSTCK!MCLK,@R0 ;CLEAR SECTOR AND CLOCK
4533
4534                                          ;NOW ALL REGISTERS WILL BE COMPARED
4535                                          ;SO FILL EXPECTED VALUE INTO SAVED LOCATIONS
4536
4537  023316  052737  100000  003142          BIS     #SC,@#REINTO+6  ;INCLUDE SC IN SAVED RHCS1
4538  023324  053737  002004  003160          BIS     @#ATTENT,@#REINTO+24 ;FILL APPROPIATE ATTENTION
4539                                                                  ;IN SAVED RHAS
4540  023332  052737  000001  003162          BIS     #DMD,@#REINTO+26 ;SET DMD IN RHMR SAVED
4541  023340  052737  100000  003164          BIS     #ATA,@#REINTO+30 ;SET ATA IN RHDS1 SAVED
4542  023346  013737  001172  003176          MOV     @#$TMP1,@#REINTO+42 ;MOVE EXPECTED VALUE
4543                                                                  ;INTO RHLA SAVED
4544
4545                                          ;AFTER SEARCH COMMAND
4546                                          ;SAVE REGISTERS AGAIN SO THAT COMPARES CAN BE DONE
4547
4548  023354  004037  041214                  JSR     R0,@#SAVER      ;SAVE
4549  023360  001624                          RHMC                    ;FROM
4550  023362  002070                          WRFROM                  ;TO
4551  023364  000023                          19.                     ;NUMBER
4552
4553                                          ;AS UPPER BYTE OF RHAS CAN BE CHANGING IN A DUAL PORT
4554                                          ;OPERATION THE UPPER BYTE OF RHAS WILL BE SAVED AS IS
4555                                          ;SO THAT THE COMPARES ARE ONLY VALID FOR THE LOWER BYTE
4556  023366  113737  003161  002115          MOVB    @#REINTO+25,@#WRFROM+25;SAVE UPPER RHAS
4557
4558
4559                                          ;COMPARE REGISTERS BEFORE SEARCH WITH AFTER
4560
4561  023374  004037  041416                  JSR     R0,@#COMPAR     ;COMPAR
4562  023400  003134                          REINTO                  ;GO BUFFER
```

```
4563
4564  023402  002070                          WRFROM                  ;TEST BUFFER
4565  023404  000022                          18.                     ;NUMBER
4566  023406  023414                          13$                     ;RETURN FOR ERROR
4567  023410  023414                          13$                     ;SAME
4568  023412  023434                          14$                     ;RETURN FOR GOOD COMPARISON
4569  023414  013705  045272          13$:    MOV     @#ERWORD,R5     ;GETTING READY TO INDEX
4570  023420  060505                          ADD     R5,R5           ;DOUBLE ERROR WORD
4571  023422  016537  001622  040172          MOV     RHWC-2(R5),@#REGADR ;FAILING REG. ADDRESS
4572  023430  104001                          ERROR   1               ;CONTENTS OF REGISTER
4573  023432  000207                          RTS     PC              ;CHANGED AT END OF
4574  023434                          14$:                            ;SEARCH
4575
```

```
4576
4577                                    ;*****************************************************************
4578                                    ;*TEST 54        MAKE CURRENT CYLINDER = 0
4579                                    ;*****************************************************************
4580  023434  000004                    TST54:  SCOPE
4581  023436  012706  001000                    MOV     #STACK,SP       ;RESET STACK
4582  023442  004737  040522                    JSR PC,@#CLDISK         ;INIT DRIVE
4583  023446  012777  000001  156176            MOV     #DMD,@RHMR      ;SET DIAGNOSTIC MODE
4584  023454  004037  043074                    JSR     R0,@#MAKECYL    ;SUBROUTINE TO GIVE A SEEK
4585                                                                    ;COMMAND FOLOWED BY A INIT
4586                                                                    ;THIS SHUOLD CHANGE RHCC
4587  023460  000000                            0                       ;CHANGE RHCC TO 0
4588
```

```
                                .SBTTL  READ/WRITE ADDRESSING VIA RHMR


                                ;*****************************************************************
                                ;*TEST 55        WRITE HEADER AND DATA 1

                                ;*      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 PASS
                                ;*      BECAUSE ERROR LOGIC HAS NOT BEEN CHECKED YET (ON FIRST PASS)

                                ;*****************************************************************
023462  000004                  TST55:  SCOPE
023464  012706  001000                  MOV     #STACK,SP       ;RESET STACK
023470  012737  000055  002020          MOV     #TTNO,@#TSTNM   ;THIS SAVES TEST NUMBER

023476  012700  046772                  MOV     #SECGAP,R0      ;POINTER
023502  012701  000460                  MOV     #304.,R1        ;COUNTER
023506  012720  177777          1$:     MOV     #-1,(R0)+       ;CLEAR "DISK" AREA TO ALL ONES
023512  005301                          DEC     R1              ;
023514  001374                          BNE     1$              ;
023516  004767  015000                  JSR     PC,CLDISK       ;THIS IS USED TO SET UP GENERAL
                                                                ;REGISTER CORRESPONDENCE

                                        ;THESE ARE TO SET UP FOR DISKLESS USE ONLY

023522  012737  010000  050274          MOV     #FMT22,@#WCYL   ;FORMAT 22=16 BIT WORDS AND
                                                                ;CYLINDER 0
023530  005037  050276                  CLR     @#WSECTR        ;TRACK=0, SECTOR=0
023534  005037  050300                  CLR     @#WKEY1         ;KEY1=0
023540  005037  050302                  CLR     @#WKEY2         ;KEY2=0
023544  012737  000400  050334          MOV     #256.,@#FNWORD  ;256 DATAWORDS
023552  004537  041730                  JSR     R5,@#CRC        ;GO TO CALCULATE CRC
023556  050274                          WCYL
023560  050304                          GCRC

                                        ;THESE ARE REGULAR SETUPS FOR RH11 & "WRFROM" OUTPUT BUFFER

023562  012777  177374  156034          MOV     #-260.,@RHWC    ;256 DATA WORDS 4 HEADER WORDS
023570  012700  002070                  MOV     #WRFROM,R0      ;BUS ADDRESS TO BE
023574  010077  156026                  MOV     R0,@RHBA        ;BUFFER "WRFROM"
023600  012705  000403                  MOV     #259.,R5        ;COUNTER
023604  012720  010000                  MOV     #FMT22,(R0)+    ;FORMAT =16 BIT WORD

023610  005020                  2$:     CLR     (R0)+           ;SECTOR=0, TRACK=0,KEYS=0, ALL DATA=0
023612  005305                          DEC     R5              ;& CYLINDER=0....SO CLEAR ALL "WRFROM"
023614  001375                          BNE     2$              ;CONTINUE IF ALL 259 NOT COMPLETE
023616  005077  156014                  CLR     @RHDST          ;TRACK=0, SECTOR=0
```

```
4643  023622  004737  040556                  JSR     PC,@#CHECKT     ;CHECK THAT DVA,RDY,DPR,DRY = 1
4644  023626  104400  057176                  TYPE    ,CPHALT         ;AND THAT NO OTHERS = 1.  CANNOT CON-
4645                                                                  ;TINUE TESTING IF BOTH AREN'T TRUE
4646  023632  000000                          HALT                    ;STOP THE TEST
4647
4648  023634  013711  002046                  MOV     @#WRIFOR,@R1    ;GET READY FOR WRITE HEADER
4649                                                                  ;AND DATA WITH 62 IN RHCS1
4650  023640  005037  001774                  CLR     @#ERFLG$        ;CLEAR ERROR FLAG
4651  023644  012777  010000  155770          MOV     #FMT22,@RHOF    ;FORMAT BIT=1 16 BIT WORDS
4652  023652  005077  155766                  CLR     @RHCA           ;CYLINDER 0
4653  023656  004737  050134                  JSR     PC,@#COM&HD     ;WRITE HEADER AND DATA FROM "WRFROM"
4654                                                                  ;INTO THE RHMR REGISTER AND BACK INTO
4655                                                                  ;CORE "DISK" AREA
4656
4657                                          ;IF THE PROGRAM COMES BACK HERE WITHOUT ERROR
4658                                          ;PRINT OUTS FROM THE "COM&HD" ROUTINE THAT MEANS
4659                                          ;ALL HEADER ON DISK IS GOOD IE. ONLY DATA IS
4660                                          ;TO BE CHECKED TO SEE IF IT IS ZERO
4661
4662                                          ;AND WRITE DATA GAP AND TOLERANCE GAP TO SEE IF THEY
4663                                          ;ARE ALL ZEROS (ECC1 AND ECC2 MAY NOT BE 0) AS WELL AS
4664                                          ;CHECKING RHWC FOR ZERO
4665
4666  023662  017767  155736  155236          MOV     @RHWC,$BDDAT    ;MOVE WORD COUNTER INTO BAD DATA
4667  023670  001401                          BEQ     5$              ;SHOULD HAVE COUNTED UP TO ZERO
4668  023672  104040                          ERROR   40              ;RHWC DID NOT = 0 AFTER A WRITE
4669                                                                  ;HEADER AND DATA
4670
4671  023674  005737  001774          5$:     TST     @#ERFLG$        ;HAVE ANY ERRORS OCCURED?
4672  023700  001034                          BNE     TST56           ;;BRANCH IF YES
4673  023702  004737  040746                  JSR     PC,@#CHECKE     ;CHECK THAT BITS = 1
4674  023706  104400  057176                  TYPE    ,CPHALT         ;CANNOT CONTINUE TESTING IF THEY DON'T
4675  023712  000000                          HALT                    ;STOP THE TEST
4676  023714  005037  050070                  CLR     @#ECC1          ;CLEAR ECC
4677  023720  005037  050072                  CLR     @#ECC2
4678
4679                                          ;REINTO BUFFER IS FILLED WITH EXPECTED DATA OF ALL 0'S
4680
4681  023724  004037  040440                  JSR     R0,@#CLAREA     ;CLEAR "REINTO"
4682  023730  003134                                  REINTO          ;FROM
4683  023732  004174                                  REINTO+<272.*2> ;TO
4684  023734  000000                          .WORD   0               ;FILL WITH ZEROS
4685  023736  005037  001774                  CLR     @#ERFLG$        ;CLEAR ERROR FLAG
4686
4687
4688                                          ;COMPARE "REINTO" BUFFER WITH "DISK" BUFFER
4689  023742  004037  041416                  JSR     R0,@#COMPAR     ;CHECK
4690  023746  003134                          REINTO                  ;GOOD BUFFER
4691  023750  047070                          DISK                    ;TEST BUFFER
4692  023752  000421                          273.                    ;NUMBER OF WORDS CHECKED
4693  023754  023762                          3$                      ;RETURN POINT FOR ERROR HEADER
4694  023756  023766                          4$                      ;RETURN POINT FOR ERROR DATA
4695  023760  023772                          TST56                   ;RETURN FOR GOOD COMPARISON
4696  023762  104007          3$:     ERROR   7               ;READ ERROR10 NEXT
```

```
 4697  023764  000207                          RTS     PC              ;RETURN TO "COMPAR"
 4698  023766  104010                  4$:     ERROR   10              ;WORD NOS 1 TO 256 ARE
 4699                                                                  ;DATA WORDS
 4700                                                                  ;257 AND 258 ARE ECC
 4701                                                                  ;ZEROED OUT
 4702                                                                  ;259 TO 273 TOLERANCE GAP
 4703  023770  000207                          RTS     PC
 4704
```

```
                                        ;*****************************************************************
                                        ;*TEST 56       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)
                                        ;
                                        ;*****************************************************************
023772  000004                  TST56:  SCOPE
023774  012706  001000                  MOV     #STACK,SP       ;RESET STACK
024000  012737  000056  002020          MOV     #TTNO,@#TSTNM   ;THIS SAVES TEST NUMBER
                                        ;
024006  012700  046772                  MOV     #SECGAP,R0      ;POINTER
024012  012701  000460                  MOV     #304.,R1        ;COUNTER
024016  005020                  1$:     CLR     (R0)+           ;CLEAR SIMULATED "DISK" AREA IN CORE
024020  005301                          DEC     R1              ;
024022  001375                          BNE     1$              ;
024024  004737  040522                  JSR     PC,@#CLDISK     ;THIS IS USED TO SET GENERAL REGISTERS
                                        ;
                                        ;THESE ARE TO BE SETUP FOR DISKLESS USE ONLY
                                        ;
024030  012737  010000  050274          MOV     #FMT22,@#WCYL   ;FORMAT 22 = 16 BIT WORDS AND
                                                                ;CYLINDER 0
024036  012737  000001  050276          MOV     #1,@#WSECTR     ;TRACK=0, SECTOR=1
024044  005037  050300                  CLR     @#WKEY1         ;KEY1=0
024050  005037  050302                  CLR     @#WKEY2         ;KEY2=0
024054  012737  000400  050334          MOV     #256.,@#FNWORD  ;256 DATA WORDS
024062  004537  041730                  JSR     R5,@#CRC        ;GO TO CALCULATE CRC
024066  050274                          WCYL
024070  050304                          GCRC
                                        ;
                                        ;THESE ARE REGULAR SETUPS FOR THE RH11 AND "WRFROM" BUFFER
                                        ;
024072  012777  177374  155524          MOV     #-260.,@RHWC    ;256 DATA WORDS 4 HEADER WORDS
024100  012700  002070                  MOV     #WRFROM,R0      ;THESE TWO INSTRUCTIONS GETS
024104  010077  155516                  MOV     R0,@RHBA        ;ADDR. OF WRFROM INTO R0 AND
                                                                ;BUS ADDRESS REGISTER
024110  012720  010000                  MOV     #FMT22,(R0)+    ;FORMAT=16 BIT WORDS
                                                                ;CYLINDER=0
024114  012720  000001          2$:     MOV     #1,(R0)+        ;TRACK=0, SECTOR=1, KEYS=0
024120  005020                          CLR     (R0)+           ;KEY1=0
024122  005020                          CLR     (R0)+           ;KEY2=0
024124  012705  000400                  MOV     #256.,R5        ;COUNTER
024130  012720  177777          3$:     MOV     #-1,(R0)+       ;MOVE ALL ONES FOR DATA
024134  005305                          DEC     R5
024136  001374                          BNE     3$              ;BRANCH IF DATA NOT COMPLETE
024140  012777  000001  155470          MOV     #1,@RHDST       ;TRACK=0 SECTOR=1
                                        ;
024146  004737  040556                  JSR     PC,@#CHECKT     ;CHECK THAT DVA,RDY,DPR,DRY = 1
024152  104400  057176                  TYPE    ,CPHALT         ;AND THAT NO OTHERS = 1.  CANNOT CON-
                                                                ;TINUE TESTING IF BOTH AREN'T TRUE
```

```
4759  024156  000000                          HALT                    ;STOP THE TEST
4760
4761  024160  013711  002046                  MOV     @#WRIFOR,@R1    ;GET READY FOR WRITE HEADER AND
4762                                                                    ;DATA WITH 62 IN RHCS1
4763  024164  005037  001774                  CLR     @#ERFLG$        ;CLEAR ERROR FLAG
4764  024170  012777  010000  155444          MOV     #FMT22,@RHOF    ;FORMAT BIT=1 (16 BIT WORDS)
4765  024176  005077  155442                  CLR     @RHCA           ;CYLINDER =0
4766  024202  004737  050134                  JSR     PC,@#COM%HD     ;WRITE HEADER AND DATA INTO "DISK" AREA
4767
4768                                          ;IF THE PROGRAM COMES BACK HERE WITHOUT ERROR PRINTOUTS
4769                                          ;FROM THE "COM%HD" ROUTINE THAT MEANS ALL THE HEADER ON "DISK"
4770                                          ;IS GOOD IE. ONLY THE DATA IS TO BE CHECKED TO SEE IF IT IS
4771                                          ;ALL ONES AND WRITE DATA GAP AND TOLERANCE GAP TO SEE IF
4772                                          ;THEY ARE ALL ZEROS, - ECC1 AND ECC2 ARE NOT CHECKED.
4773
4774                                          ;RHWC IS CHECKED TO BE = 0
4775
4776  024206  017767  155412  154712          MOV     @RHWC,$BDDAT    ;LOAD WORD COUNTER JUST IN CASE
4777  024214  001401                          BEQ     6$              ;SHOULD BE = 0
4778  024216  104040                          ERROR   40              ;RHWC DOES NOT = 0 AFTER A WRITE
4779                                                                  ;HEADER AND DATA IS COMPLETED
4780
4781  024220  005737  001774          6$:     TST     @#ERFLG$        ;HAVE ANY ERRORS OCCURRED?
4782  024224  001041                          BNE     TST57           ;BRANCH IF YES
4783  024226  004737  040746                  JSR     PC,@#CHECKE     ;CHECK THAT BITS = 1
4784  024232  104400  057176                  TYPE    ,CPHALT         ;CANNOT CONTINUE TESTING IF THEY DON'T
4785  024236  000000                          HALT                    ;STOP THE TEST
4786  024240  005037  050070                  CLR     @#WECC1         ;CLEAR ECC
4787  024244  005037  050072                  CLR     @#WECC2         ;CLEAR ECC
4788
4789                                          ;FILL "REINTO" BUFFER WITH EXPECTED DATA OF ALL 1'S
4790
4791  024250  004037  040440                  JSR     R0,@#CLAREA     ;FILL REINTO BUFFER
4792  024254  003134                                  REINTO          ;FROM
4793  024256  004132                                  REINTO+<255.*2> ;TO
4794  024260  177777                          .WORD   -1              ;DATA
4795  024262  004037  040440                  JSR     R0,@#CLAREA     ;FILL REST
4796  024266  004134                          REINTO+<256.*2>         ;FROM
4797  024270  004174                          REINTO+<272.*2>         ;TO
4798  024272  000000                          0                       ;DATA
4799
4800  024274  005037  001774                  CLR     @#ERFLG$        ;CLEAR ERROR FLAG
4801
4802                                          ;NOW COMPARE "DISK" BUFFER WITH "REINTO" BUFFER IN CORE
4803
4804  024300  004037  041416                  JSR     R0,@#COMPAR     ;CHECK
4805  024304  003134                          REINTO                  ;GOOD BUFFER
4806  024306  047070                          DISK                    ;TEST BUFFER
4807  024310  000421                          273.                    ;NUMBER OF WORDS CHECKED
4808  024312  024320                          4$                      ;RETURN POINT FOR ERROR HEADER
4809  024314  024324                          5$                      ;RETURN POINT FOR ERROR DATA
4810  024316  024330                          TST57                   ;RETURN FOR GOOD COMPARISON
4811  024320  104007                  4$:     ERROR   7               ;READ ERROR 10 NEXT
4812  024322  000207                          RTS     PC              ;RETURN TO COMPARE
```

```
4813  024324  104010                  5$:     ERROR   10          ;WORD NOS 1 TO 256 ARE
4814                                                              ;DATA WORDS
4815                                                              ;WORD NOS 257 AND 258
4816                                                              ;ARE ECC WHICH HAVE BEEN
4817                                                              ;ZEROED
4818                                                              ;WORD NOS 259
4819                                                              ;IS DATA GAP
4820                                                              ;WORD NOS 260 TO 273
4821                                                              ;ARE TOLERANCE GAP
4822  024326  000207                          RTS     PC          ;RETURN TO COMPARE
4823
4824
4825
```

```
                                ;**************************************************************
                                ;*TEST 57       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)
                                ;**************************************************************
024330  000004                TST57:  SCOPE
024332  012706  001000                MOV     #STACK,SP       ;RESET STACK
024336  012737  000057  002020        MOV     #TTNO,@#TSTNM   ;THIS SAVES TEST NUMBER
024344  012700  046772                MOV     #SECGAP,R0      ;POINTER
024350  012701  000460                MOV     #304.,R1        ;COUNTER
024354  012720  000377          1$:   MOV     #377,(R0)+      ;LOAD SIMULATED "DISK" AREA WITH 377
024360  005301                        DEC     R1              ;
024362  001374                        BNE     1$              ;
024364  004767  014132                JSR     PC,CLDISK       ;THIS IS USED TO SET UP GENERAL
                                                              ;REGISTER CORRESPONDENCE

                                ;THESE ARE TO BE SETUP FOR DISKLESS USE ONLY

024370  012737  010000  050274        MOV     #FMT22,@#WCYL   ;FORMAT 22=16 BIT WORDS AND
                                                              ;CYLINDER 0
024376  012737  000401  050276        MOV     #401,@#WSECTR   ;TRACK=1, SECTOR=1
024404  005037  050300                CLR     @#WKEY1         ;KEY1=0
024410  005037  050302                CLR     @#WKEY2         ;KEY2=0
024414  012737  000400  050334        MOV     #256.,@#FNWORD  ;256 DATA WORDS
024422  004537  041730                JSR     R5,@#CRC        ;GO TO CALCULATE CRC
024426  050274                        WCYL
024430  050304                        GCRC

                                ;THESE ARE REGULAR SETUPS FOR RH11 AND "WRFROM" BUFFER

024432  012777  177374  155164        MOV     #-260.,@RHWC    ;256 DATA WORDS 4 HEADER WORDS
024440  012700  002070                MOV     #WRFROM,R0      ;THESE TWO INSTRUCTIONS GET
                                                              ;ADDR. OF WRFROM INTO R0
024444  010077  155156                MOV     R0,@RHBA        ;AND BUS ADDRESS REGISTER

024450  012720  010000                MOV     #FMT22,(R0)+    ;FORMAT=16 BIT WORDS
                                                              ;CYLINDER=0
024454  012720  000401          2$:   MOV     #401,(R0)+      ;TRACK=1, SECTOR=1, KEYS=0
024460  005020                        CLR     (R0)+           ;KEY1=0
024462  005020                        CLR     (R0)+           ;KEY2=0
024464  012705  000400                MOV     #256.,R5        ;COUNTER
024470  012720  052525          3$:   MOV     #052525,(R0)+   ;MOVE ALTERNATE ONES FOR DATA
024474  005305                        DEC     R5              ;COUNT
024476  001374                        BNE     3$              ;BRANCH IF DATA NOT COMPLETE
024500  012777  000401  155130        MOV     #401,@RHDST     ;TRACK=1 SECTOR=1

024506  004737  040556                JSR     PC,@#CHECKT     ;CHECK THAT DVA,RDY,DPR,DRY = 1
024512  104400  057176                TYPE    ,CPHALT         ;AND THAT NO OTHERS = 1.  CANNOT CON-
```

```
                                                        ;TINUE TESTING IF BOTH AREN'T TRUE
024516  000000                  HALT                    ;STOP THE TEST

024520  013711  002046          MOV     @#WRIFOR,@R1    ;GET READY FOR WRITE HEADER
                                                        ;AND DATA WITH 62 IN RHCS1
024524  005037  001774          CLR     @#ERFLGS        ;CLEAR ERROR FLAG
024530  012777  010000  155104  MOV     #FMT22,@RHOF    ;FORMAT BIT=1(16 BIT WORDS
024536  005077  155102          CLR     @RHCA           ;CYLINDER=0
024542  004737  050134          JSR     PC,@#COMWHD     ;WRITE HEADER AND DATA INTO "DISK" CORE

                                ;IF THE PROGRAM COMES BACK HERE WITHOUT ERROR PRINTOUTS
                                ;FROM THE "COMWHD" ROUTINE THAT MEANS ALL HEADER IN
                                ;"DISK" IS GOOD.  DATA IS TO BE CHECKED TO SEE
                                ;IF IT IS ALL 052525 AND WRITE DATA GAP AND
                                ;TOLERANCE GAP TO SEE IF THEY ARE ALL ZEROS.

                                ;RHWC IS CHECKED TO BE = 0

024546  017767  155052  154352  MOV     @RHWC,$3DDAT    ;LOAD AND TEST FOR ZERO
024554  001401                  BEQ     6$              ;RHWC SHOULD = 0
024556  104040                  ERROR   40              ;RHWC DID NOT = 0 AFTER A WRITE
                                                        ;HEADER AND DATA WAS COMPLETED

                                ;ONLY ECC1 AND ECC2 ARE NOT CHECKED

024560  005737  001774  6$:     TST     @#ERFLGS        ;HAVE ANY ERRORS OCCURED?
024564  001041                  BNE     TST60           ;;BRANCH IF YES
024566  004737  040746          JSR     PC,@#CHECKE     ;CHECK THAT BITS = 1
024572  104400  057176          TYPE    ,CPHALT         ;CANNOT CONTINUE TESTING IF THEY DON'T
024576  000000                  HALT                    ;STOP THE TEST
024600  005037  050070          CLR     @#WECC1         ;CLEAR ECC
024604  005037  050072          CLR     @#WECC2         ;CLEAR ECC

                                ;FILL "REINTO" BUFFER WITH EXPECTED DATA

024610  004037  040440          JSR     R0,@#CLAREA     ;FILL REINTO BUFFER
024614  003134                  REINTO                  ;FROM
024616  004132                  REINTO+<255.*2>         ;TO
024620  052525                  .WORD   52525           ;DATA
024622  004037  040440          JSR     R0,@#CLAREA     ;FILL REST
024626  004134                          REINTO+<256.*2> ;FROM
024630  004174                  REINTO+<272.*2>         ;TO
024632  000000                  .WORD   0               ;DATA
024634  005037  001774          CLR     @#ERFLGS        ;CLEAR ERROR FLAG

                                ;NOW COMPARE "DISK" BUFFER WITH "REINTO" BUFFER IN CORE

024640  004037  041416          JSR     R0,@#COMPAR     ;CHECK
024644  003134                  REINTO                  ;GOOD BUFFER
024646  047070                  DISK                    ;TEST BUFFER
024650  000421                  273.                    ;NUMBER OF WORDS CHECKED
024652  024660                  4$                      ;RETURN POINT FOR ERROR HEADER
024654  024664                  5$                      ;RETURN POINT FOR ERROR DATA
024656  024670                  TST60                   ;RETURN FOR GOOD COMPARISON
```

```
4934  024660  104007          45:   ERROR   7       ;READ ERROR 10 NEXT
4935  024662  000207                RTS     PC      ;RETURN TO COMPARE
4936  024664  104010          55:   ERROR   10      ;WORD NOS 1 TO 256 ARE
4937                                                ;DATA WORDS
4938                                                ;WORD NOS 257 AND 258
4939                                                ;ARE ECC WHICH HAVE BEEN
4940                                                ;ZEROED
4941                                                ;WORD NOS 259
4942                                                ;IS DATA GAP
4943                                                ;WORD NOS 260 TO 273
4944                                                ;ARE TOLERANCE GAP
4945  024666  000207                RTS     PC      ;RETURN TO COMPARE
4946
4947
4948
4949
```

```
                                ;*****************************************************************
                                ;*TEST 60       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
                                ;
                                ;*****************************************************************
024670  000004                TST60:  SCOPE
024672  012706  001000                MOV     #STACK,SP       ;RESET STACK
024676  012737  000060  002020        MOV     #TTNO,@#TSTNM   ;THIS SAVES TEST NUMBER
                                ;
024704  012700  046772                MOV     #SECGAP,R0      ;POINTER
024710  012701  000460                MOV     #304.,R1        ;COUNTER
024714  012720  177777        1$:     MOV     #-1,(R0)+       ;CLEAR DISK AREA TO ALL ONES.
024720  005301                        DEC     R1              ;
024722  001374                        BNE     1$              ;
024724  004767  013572                JSR     PC,CLDISK       ;THIS IS USED TO SET GENERAL
                                                                ;REGISTERS
                                ;
                                        ;THESE ARE TO SET UP FOR DISKLESS USE ONLY
                                ;
024730  012737  010000  050274        MOV     #FMT22,@#WCYL   ;FORMAT 22=16 BITWORDS AND
                                                                ;CYLINDER 0
024736  005037  050276                CLR     @#WSECTR        ;TRACK=0, SECTOR=0
024742  005037  050300                CLR     @#WKEY1         ;KEY1=0
024746  005037  050302                CLR     @#WKEY2         ;KEY2=0
024752  012737  000400  050334        MOV     #256.,@#FNWORD  ;256 DATAWORDS
024760  004537  041730                JSR     R5,@#CRC        ;GO TO CALCULATE CRC
024764  050274                        WCYL
024766  050304                        GCRC
                                ;
                                        ;THESE ARE REGULAR SETUPS
                                ;
024770  012777  177374  154626        MOV     #-260.,@RHWC    ;256 DATA WORDS 4 HEADER WORDS
024776  012700  002070                MOV     #WRFROM,R0      ;FROM BUFFER "WRFROM"
025002  010077  154620                MOV     R0,@RHBA        ;IN BUS ADDRESS
025006  012705  000403                MOV     #259.,R5        ;COUNTER
025012  012720  010000                MOV     #FMT22,(R0)+    ;FORMAT =16 BIT WORD
                                                                ;CYLINDER=0
025016  005020                2$:     CLR     (R0)+           ;SECTOR=0, TRACK=0,KEYS=0, ALL DATA=0
025020  005305                        DEC     R5              ;COUNT
025022  001375                        BNE     2$              ;BRANCH IF ALL 259 NOT COMPLETE
025024  005077  174606                CLR     @RHDST          ;TRACK=0, SECTOR=0
                                ;
025030  004737  040556                JSR     PC,@#CHECKT     ;CHECK THAT DVA,RDY,DPR,DRY = 1
025034  104400  057176                TYPE    ,CPHALT         ;AND THAT NO OTHERS = 1.  CANNOT CON-
                                                                ;TINUE TESTING IF BOTH AREN'T TRUE
025040  000000                        HALT                    ;STOP THE TEST
                                ;
025042  013711  002046                MOV     @#WRIFOR,@R1    ;GET READY FOR WRITE HEADER
                                                                ;AND DATA WITH 62 IN RHCS1
```

```
5004  025046  005037  001774                CLR     @#ERFLG$        ;CLEAR ERROR FLAG
5005  025052  012777  010000  154562        MOV     #FMT22,@RHOF    ;FORMAT BIT=1 16 BIT WORDS
5006  025060  005077  154560                CLR     @RHCA           ;CYLINDER 0
5007  025064  012777  000001  154560        MOV     #DMD,@RHMR      ;SET DIAGNOSTIC MODE
5008  025072  052777  000001  154532        BIS     #GO,@RHCS1      ;GO
5009  025100  000240                        NOP
5010  025102  052777  000001  154522        BIS     #GO,@RHCS1      ;THIS GO SHOULD SET PGE
5011
5012  025110  004737  040126                JSR     PC,@#PUTREG     ;SAVE REGISTERS
5013  025114  032737  002000  001702        BIT     #PGE,@#CS1      ;IS PGE SET
5014  025122  001404                        BEQ     3$              ;BRANCH IF GOOD
5015  025124  013737  001630  001122        MOV     @#RHCS2,@#SBDADR
5016  025132  104037                        ERROR   37              ;PGE DID NOT SET WHEN A WRITE
5017                                                                ;WAS ATTEMPTED WITH ONE IN PROGRESS
5018  025134                          3$:
```

```
5019
5020                            ;*****************************************************************
5021                            ;*      THESE TESTS ARE THROUGH THE MAINTAINABILITY REGISTER - RHMR
5022                            ;*
5023                            ;*      THE SECTOR GAP AND SYNC BYTE ARE ALWAYS READ AS
5024                            ;*      ZEROS AND 144000 NO MATTER WHAT IS IN THE SIMULATED DISK AREA
5025                            ;*      TAGGED SECGAP: AND WSSYNC:
5026                            ;*
5027                            ;*      THE HEADER CONSISTING OF CYLINDER ADDRESS, SECTOR,
5028                            ;*      TRACK AND THE KEYS ARE READ FROM LOCATION
5029                            ;*      CYL:, SECTOR:, KEY1:, AND KEY2 AND NOT FROM
5030                            ;*      HEADER: ON SIMULATED DISK
5031                            ;*
5032                            ;*      CRC IS READ FROM SIMULATED DISK LOCATION WCRC:
5033                            ;*      HEADER GAP IS ALWAYS READ AS ZEROS NO MATTER
5034                            ;*      WHAT IS ON THE SIMULATED DISK AREA
5035                            ;*
5036                            ;*      THE DATA SYNC IS READ  FROM HDWSYN:
5037                            ;*      ON SIMULATED DISK
5038                            ;*
5039                            ;*      ALL DATA IS READ FROM SIMULATED DISK DISK:
5040                            ;*****************************************************************
```

```
                                        ;*****************************************************************
                                        ;*TEST 61       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)
                                        ;*****************************************************************
025134  000004                  TST61:  SCOPE
025136  012706  001000                  MOV     #STACK,SP       ;RESET STACK
025142  012737  000061  002020          MOV     #TTNO,@#TSTNM   ;THIS SAVES TEST NUMBER

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

025150  012746  000000                  MOV     #0,-(SP)        ;DATA TO BE READ
025154  012705  000400                  MOV     #256.,R5        ;COUNTER
025160  012700  047070                  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  045152          MOV     #0!FMT22,@#CYL ;16 BITS PER WORD
                                                                ;CYLINDER 0, FORMAT 16 BITS
025214  112737  000000  045155          MOVB    #0,@#SECOTR+1  ;TRACK 0
025222  112737  000000  045154          MOVB    #0,@#SECOTR     ;SECTOR 0
025230  012737  000000  045156          MOV     #0,@#KEY1       ;KEY1=0
025236  012737  000000  045160          MOV     #0,@#KEY2       ;KEY2=0
025244  012737  000400  045232          MOV     #256.,@#DA#WORD ;NO. OF DATA WORDS
025252  005037  045162                  CLR     @#X             ;THIS IS A READ COMMAND
025256  004537  041730                  JSR     R5,@#CRC        ;GO TO CALCULATE CRC
025262  045152                          CYL
025264  047052                          WCRC


                                        ;THESE ARE REGULAR SETUPS FOR RH11 AND "REINTO"

025266  004737  040522                  JSR     PC,@#CLDISK     ;SETUP GENERAL REGISTERS
025272  012777  177374  154324          MOV     #-256.-4.,@RH#WC ;256. DATA 4 HEADER WORDS
025300  012777  003134  154320          MOV     #REINTO,@RH#BA  ;STARTING ADDRESS OF READ BUFFER
```

```
5095  025306  112746  000000                  MOVB    #0,     -(SP)   ;IN LOWER BYTE GET SECTOR
5096  025312  112766  000000  000001          MOVB    #0,     1(SP)   ;GET TRACK IN HIGHER BYTE
5097  025320  012677  154312                  MOV     (SP)+,  @RHDST  ;TRACK/SECTOR IN RHDST
5098  025324  012777  014000  154310          MOV     #FMT22!ECI,@RHOF ;16 BITS PER WORD
5099                                                                  ;ECC CORRECTION INHIBIT
5100                                                                  ;BECAUSE ECC IS NOT GOING
5101                                                                  ;TO BE CHECKED
5102  025332  005077  154306                  CLR     @RHCA           ;CYLINDER 0
5103
5104  025336  004737  040556                  JSR     PC,@#CHECKT     ;CHECK THAT DVA,RDY,DPR,DRY = 1
5105  025342  104400  057176                  TYPE    ,CPHALT         ;AND THAT NO OTHERS = 1.  CANNOT CON-
5106                                                                  ;TINUE TESTING IF BOTH AREN'T TRUE
5107  025346  000000                          HALT                    ;STOP THE TEST
5108
5109  025350  013711  002052                  MOV     @#REFOR,@R1     ;READ HEADER AND DATA=72
5110  025354  005037  001774                  CLR     @#ERFLG$        ;CLEAR ERROR FLAG
5111  025360  004737  045026                  JSR     PC,@#COMHD      ;READ HEADER AND DATA
5112
5113
5114                                          ;IF THE PROGRAM COMES BACK HERE WITHOUT ERROR PRINTOUTS
5115                                          ;FROM THE "COMHD" ROUTINE THAT MEANS SECTOR GAP,
5116                                          ;FIRST SYNC, HEADER, HEADER CRC, HEADER GAP AND
5117                                          ;SYNC BYTE HAVE GONE BY AND SYNCS WERE CORRECTLY
5118                                          ;DETECTED.
5119
5120
5121                                          ;RHWC IS CHECKED TO BE = 0 AFTER THE READ OPERATION
5122
5123  025364  017767  154234  153534          MOV     @RHWC,$BDDAT    ;LOAD AND TEST RHWC
5124  025372  001401                          BEQ     20$             ;SHOULD = 0
5125  025374  104040                          ERROR   40              ;RHWC DOES NOT = 0 AFTER A READ
5126                                                                  ;HEADER AND DATA
5127
5128                                          ;HEADER AND DATA ARE TO BE CHECKED.
5129                                          ;IN CHECKING READ DATA THE WRITE FROM BUFFER
5130                                          ;"WRFROM" IS FILLED WITH EXPECTED DATA AND
5131                                          ;COMPARISONS ARE MADE.
5132
5133  025376  005737  001774          20$:    TST     @#ERFLG$        ;ANY ERRORS ALREADY THERE
5134  025402  001046                          BNE     TST62           ;BRANCH IF YES
5135  025404  004737  040746                  JSR     PC,@#CHECKE     ;CHECK THAT BITS = 1
5136  025410  104400  057176                  TYPE    ,CPHALT         ;CANNOT CONTINUE TESTING IF THEY DON'T
5137  025414  000000                          HALT                    ;STOP THE TEST
5138  025416  012700  002070                  MOV     #WRFROM,R0      ;GETTING READY TO FILL EXPECTED DATA
5139  025422  012720  010000                  MOV     #0!FMT22,(R0)+  ;CYLINDER 0
5140  025426  112746  000000                  MOVB    #0,-(SP)        ;IN LOWER BYTE GET SECTOR
5141  025432  112766  000000  000001          MOVB    #0,1(SP)        ;GET TRACK IN HIGHER BYTE
5142  025440  012620                          MOV     (SP)+,(R0)+     ;GET TRACK/SECTOR IN BUFFER
5143  025442  012720  000000                  MOV     #0,(R0)+        ;KEY1 IN BUFFER
5144  025446  012720  000000                  MOV     #0,(R0)+        ;KEY2 IN BUFFER
5145  025452  012701  000400                  MOV     #256.,R1        ;DATA WORD COUNTER
5146  025456  012702  000000                  MOV     #0,R2           ;DATA
5147  025462  010220                  3$:     MOV     R2,(R0)+        ;DATA INTO BUFFER
5148  025464  005301                          DEC     R1              ;COUNT
```

```
5149  025466  001375                          BNE     3$              ;BRANCH IF 256 NOT DONE
5150
5151                                          ;NOW READ DATA BUFFER WILL BE CHECKED
5152
5153  025470  004037  041416                  JSR     R0,@#COMPAR     ;CHECK
5154  025474  002070                          WRFROM                  ;GOOD BUFFER
5155  025476  003134                          REINTO                  ;TEST BUFFER
5156  025500  000404                          4+256.                  ;NUMBER OF WORDS CHECKED
5157  025502  025510                          4$                      ;RETURN POINT FOR ERROR HEADER
5158  025504  025514                          5$                      ;RETURN POINT FOR ERROR DATA
5159  025506  025520                          TST62                   ;RETURN FOR GOOD COMPARISON
5160  025510  104004                  4$:     ERROR   4               ;READ NEXT ERROR
5161  025512  000207                          RTS     PC              ;RETURN TO "COMPAR"
5162  025514  104005                  5$:     ERROR   5               ;WORD NOS 1 TO 4 ARE
5163                                                                  ;HEADER WORDS
5164                                                                  ;5 TO 260 ARE DATA WORDS
5165  025516  000207                          RTS     PC              ;RETURN TO "COMPAR"
5166
5167
5168
5169
5170
```

```
5171
5172                                    ;*****************************************************************
5173                                    ;*TEST 62       READ HEADER AND DATA 2
5174                                    ;*      READ CYLINDER 0 FORMAT 16 BITS PER WORD
5175                                    ;*      TRACK 0, SECTOR 1, KEYS 0, 256 WORDS OF 177777
5176                                    ;*      ANY ERROR LOGIC INDICATIONS ARE NOT CONCLUSIVE ON FIRST PASS
5177                                    ;*      BECAUSE ERROR LOGIC HAS NOT BEEN CHECKED YET (ON FIRST PASS)
5178
5179                                    ;*****************************************************************
5180  025520  000004                    TST62:  SCOPE
5181  025522  012706  001000                    MOV     #STACK,SP       ;RESET STACK
5182  025526  012737  000062  002020            MOV     #TTNO,@#TSTNM   ;THIS SAVES TEST NUMBER
5183
5184
5185                                    ;       SETUP FOR WHAT IS TO BE READ
5186                                    ;       HEADER CRC IS RESTORED FROM A SUBROUTINE
5187
5188  025534  012746  177777                    MOV     #-1,-(SP)       ;DATA TO BE READ
5189  025540  012705  000400                    MOV     #256.,R5        ;COUNTER
5190  025544  012700  047070                    MOV     #DISK,R0        ;START OF SIMULATED DISK DATA
5191  025550  011620                    1$:     MOV     (SP),(R0)+      ;MOVE IN DATA ON TO SIMULATED DISK
5192  025552  005305                            DEC     R5              ;COUNT
5193  025554  001375                            BNE     1$              ;BRANCH IF 256 NOT COMPLETE
5194  025556  005726                            TST     (SP)+           ;UNDO -(SP)
5195  025560  012705  000021                    MOV     #17.,R5         ;2 ECC WORDS
5196                                                                    ;1 DATA GAP
5197                                                                    ;14 TOLERANCE GAP
5198  025564  005020                    2$:     CLR     (R0)+           ;CLEAR ECC, DATA GAP, AND
5199  025566  005305                            DEC     R5              ;TOLERANCE GAP
5200  025570  001375                            BNE     2$              ;BRANCH IF NOT COMPLETE
5201
5202
5203                                            ;THESE ARE TO SETUP FOR DISKLESS USE ONLY
5204
5205  025572  012737  010000  045152            MOV     #0!FMT22,@#CYL ;16 BITS PER WORD
5206                                                                    ;CYLINDER 0, FORMAT 16 BITS
5207  025600  112737  000000  045155            MOVB    #0,@#SECOTR+1  ;TRACK 0
5208  025606  112737  000001  045154            MOVB    #1,@#SECOTR     ;SECTOR 1
5209  025614  012737  000000  045156            MOV     #0,@#KEY1       ;KEY1=0
5210  025622  012737  000000  045160            MOV     #0,@#KEY2       ;KEY2=0
5211  025630  012737  000400  045232            MOV     #256.,@#DAWORD  ;NO. OF DATA WORDS
5212  025636  005037  045162                    CLR     @#X             ;THIS IS A READ COMMAND
5213  025642  004537  041730                    JSR     R5,@#CRC        ;GO TO CALCULATE CRC
5214  025646  045152                            CYL
5215  025650  047052                            WCRC
5216
5217                                            ;THESE ARE REGULAR SETUPS FOR RH11 AND "REINTO"
5218
5219  025652  004737  040522                    JSR     PC,@#CLDISK     ;SETUP GENERAL REGISTERS
5220  025656  012777  177374  153740            MOV     #-256.-4.,@RHWC ;256. DATA 4 HEADER WORDS
5221  025664  012777  003134  153734            MOV     #REINTO,@RHBA   ;STARTING ADDRESS OF READ BUFFER
5222  025672  112746  000001                    MOVB    #1,     -(SP)   ;IN LOWER BYTE GET SECTOR
5223  025676  112766  000000  000001            MOVB    #0,     1(SP)   ;GET TRACK IN HIGHER BYTE
5224  025704  012677  153726                    MOV     (SP)+,  @RHDST  ;TRACK/SECTOR IN RHDST
```

```
5225  025710  012777  014000  153724          MOV     #FMT22!ECI,@RHOF ;16 BITS PER WORD
5226                                                                    ;ECC CORRECTION INHIBIT
5227                                                                    ;BECAUSE ECC IS NOT GOING
5228                                                                    ;TO BE CHECKED
5229  025716  005077  153722                  CLR     @RHCA            ;CYLINDER 0
5230
5231  025722  004737  040556                  JSR     PC,@#CHECKT      ;CHECK THAT DVA,RDY,DPR,DRY = 1
5232  025726  104400  057176                  TYPE    ,CPHALT          ;AND THAT NO OTHERS = 1.  CANNOT CON-
5233                                                                    ;TINUE TESTING IF BOTH AREN'T TRUE
5234  025732  000000                          HALT                     ;STOP THE TEST
5235
5236  025734  013711  002052                  MOV     @#REFOR,@R1      ;READ HEADER AND DATA=72
5237  025740  005037  001774                  CLR     @#ERFLG$         ;CLEAR ERROR FLAG
5238  025744  004737  045026                  JSR     PC,@#COMHD       ;READ HEADER AND DATA
5239
5240
5241                                          ;IF THE PROGRAM COMES BACK HERE WITHOUT ERROR PRINTOUTS
5242                                          ;FROM THE "COMHD" ROUTINE THAT MEANS SECTOR GAP,
5243                                          ;FIRST SYNC, HEADER, HEADER CRC, HEADER GAP AND
5244                                          ;SYNC BYTE HAVE GONE BY AND SYNCS WERE CORRECTLY
5245                                          ;DETECTED.
5246
5247
5248                                          ;RHWC IS CHECKED TO BE = 0 AFTER THE READ OPERATION
5249
5250  025750  017767  153650  153150          MOV     @RHWC,$BDDAT     ;LOAD AND TEST RHWC
5251  025756  001401                          BEQ     20$              ;SHOULD = 0
5252  025760  104040                          ERROR   40               ;RHWC DOES NOT = 0 AFTER A READ
5253                                                                    ;HEADER AND DATA
5254
5255                                          ;HEADER AND DATA ARE TO BE CHECKED.
5256                                          ;IN CHECKING READ DATA THE WRITE FROM BUFFER
5257                                          ;"WRFROM" IS FILLED WITH EXPECTED DATA AND
5258                                          ;COMPARISONS ARE MADE.
5259
5260  025762  005737  001774          20$:    TST     @#ERFLG$         ;ANY ERRORS ALREADY THERE
5261  025766  001046                          BNE     TST63            ;BRANCH IF YES
5262  025770  004737  040746                  JSR     PC,@#CHECKE      ;CHECK THAT BITS = 1
5263  025774  104400  057176                  TYPE    ,CPHALT          ;CANNOT CONTINUE TESTING IF THEY DON'T
5264  026000  000000                          HALT                     ;STOP THE TEST
5265  026002  012700  002070                  MOV     #WRFROM,R0       ;GETTING READY TO FILL EXPECTED DATA
5266  026006  012720  010000                  MOV     #0!FMT22,(R0)+   ;CYLINDER 0
5267  026012  112746  000001                  MOVB    #1,-(SP)         ;IN LOWER BYTE GET SECTOR
5268  026016  112766  000000  000001          MOVB    #0,1(SP)         ;GET TRACK IN HIGHER BYTE
5269  026024  012620                          MOV     (SP)+,(R0)+      ;GET TRACK/SECTOR IN BUFFER
5270  026026  012720  000000                  MOV     #0,(R0)+         ;KEY1 IN BUFFER
5271  026032  012720  000000                  MOV     #0,(R0)+         ;KEY2 IN BUFFER
5272  026036  012701  000400                  MOV     #256.,R1         ;DATA WORD COUNTER
5273  026042  012702  177777                  MOV     #-1,R2           ;DATA
5274  026046  010220                  3$:     MOV     R2,(R0)+         ;DATA INTO BUFFER
5275  026050  005301                          DEC     R1               ;COUNT
5276  026052  001375                          BNE     3$               ;BRANCH IF 256 NOT DONE
5277
5278                                          ;NOW READ DATA BUFFER WILL BE CHECKED
```

```
5279
5280  026054  004037  041416                  JSR     R0,@#COMPAR     ;CHECK
5281  026060  002070                          WRFROM                  ;GOOD BUFFER
5282  026062  003134                          REINTO                  ;TEST BUFFER
5283  026064  000404                          4+256.                  ;NUMBER OF WORDS CHECKED
5284  026066  026074                          4$                      ;RETURN POINT FOR ERROR HEADER
5285  026070  026100                          5$                      ;RETURN POINT FOR ERROR DATA
5286  026072  026104                          TST63                   ;RETURN FOR GOOD COMPARISON
5287  026074  104004                  4$:     ERROR   4               ;READ NEXT ERROR
5288  026076  000207                          RTS     PC              ;RETURN TO "COMPAR"
5289  026100  104005                  5$:     ERROR   5               ;WORD NOS 1 TO 4 ARE
5290                                                                  ;HEADER WORDS
5291                                                                  ;5 TO 260 ARE DATA WORDS
5292  026102  000207                          RTS     PC              ;RETURN TO "COMPAR"
5293
5294
5295
5296
```

```
5297
5298                                    ;*****************************************************************
5299                                    ;*TEST 63       READ HEADER AND DATA 3
5300                                    ;*      READ CYLINDER 0 FORMAT 16 BITS PER WORD
5301                                    ;*      TRACK 1, SECTOR 1, KEYS 0, 256 WORDS OF 052525
5302                                    ;*      ANY ERROR LOGIC INDICATIONS ARE NOT CONCLUSIVE ON FIRST PASS
5303                                    ;*      BECAUSE ERROR LOGIC HAS NOT BEEN CHECKED YET (ON FIRST PASS)
5304
5305                                    ;*****************************************************************
5306  026104  000004                   TST63:  SCOPE
5307  026106  012706  001000                   MOV     #STACK,SP       ;RESET STACK
5308  026112  012737  000063  002020           MOV     #TTNO,@#TSTNM   ;THIS SAVES TEST NUMBER
5309
5310
5311                                    ;       SETUP FOR WHAT IS TO BE READ
5312                                    ;       HEADER CRC IS RESTORED FROM A SUBROUTINE
5313
5314  026120  012746  052525                   MOV     #052525,-(SP)   ;DATA TO BE READ
5315  026124  012705  000400                   MOV     #256.,R5        ;COUNTER
5316  026130  012700  047070                   MOV     #DISK,R0        ;START OF SIMULATED DISK DATA
5317  026134  011620                   1$:     MOV     (SP),(R0)+      ;MOVE IN DATA ON TO SIMULATED DISK
5318  026136  005305                           DEC     R5              ;COUNT
5319  026140  001375                           BNE     1$              ;BRANCH IF 256 NOT COMPLETE
5320  026142  005726                           TST     (SP)+           ;UNDO -(SP)
5321  026144  012705  000021                   MOV     #17.,R5         ;2 ECC WORDS
5322                                                                   ;1 DATA GAP
5323                                                                   ;14 TOLERANCE GAP
5324  026150  005020                   2$:     CLR     (R0)+           ;CLEAR ECC, DATA GAP, AND
5325  026152  005305                           DEC     R5              ;TOLERANCE GAP
5326  026154  001375                           BNE     2$              ;BRANCH IF NOT COMPLETE
5327
5328
5329                                    ;THESE ARE TO SETUP FOR DISKLESS USE ONLY
5330
5331  026156  012737  010000  045152           MOV     #0!FMT22,@#CYL ;16 BITS PER WORD
5332                                                                   ;CYLINDER 0, FORMAT 16 BITS
5333  026164  112737  000001  045155           MOVB    #1,@#SECOTR+1  ;TRACK 1
5334  026172  112737  000001  045154           MOVB    #1,@#SECOTR     ;SECTOR 1
5335  026200  012737  000000  045156           MOV     #0,@#KEY1       ;KEY1=0
5336  026206  012737  000000  045160           MOV     #0,@#KEY2       ;KEY2=0
5337  026214  012737  000400  045232           MOV     #256.,@#DAWORD  ;NO. OF DATA WORDS
5338  026222  005037  045162                   CLR     @#X             ;THIS IS A READ COMMAND
5339  026226  004537  041730                   JSR     R5,@#CRC        ;GO TO CALCULATE CRC
5340  026232  045152                           CYL
5341  026234  047052                           WCRC
5342
5343                                    ;THESE ARE REGULAR SETUPS FOR RH11 AND "REINTO"
5344
5345  026236  004737  040522                   JSR     PC,@#CLDISK     ;SETUP GENERAL REGISTERS
5346  026242  012777  177374  153354           MOV     #-256.-4.,@RHWC ;256. DATA 4 HEADER WORDS
5347  026250  012777  003134  153350           MOV     #REINTO,@RHBA   ;STARTING ADDRESS OF READ BUFFER
5348  026256  112746  000001                   MOVB    #1,     -(SP)   ;IN LOWER BYTE GET SECTOR
5349  026262  112766  000001  000001           MOVB    #1,     1(SP)   ;GET TRACK IN HIGHER BYTE
5350  026270  012677  153342                   MOV     (SP)+,  @RHDST  ;TRACK/SECTOR IN RHDST
```

```
5351  026274  012777  014000  153340          MOV     #FMT22!ECI,@RHOF ;16 BITS PER WORD
5352                                                                    ;ECC CORRECTION INHIBIT
5353                                                                    ;BECAUSE ECC IS NOT GOING
5354                                                                    ;TO BE CHECKED
5355  026302  005077  153336                  CLR     @RHCA            ;CYLINDER 0
5356
5357  026306  004737  040556                  JSR     PC,@#CHECKT      ;CHECK THAT DVA,RDY,DPR,DRY = 1
5358  026312  104400  057176                  TYPE    ,CPHALT          ;AND THAT NO OTHERS = 1.  CANNOT CON-
5359                                                                    ;TINUE TESTING IF BOTH AREN'T TRUE
5360  026316  000000                          HALT                     ;STOP THE TEST
5361
5362  026320  013711  002052                  MOV     @#REFOR,@R1      ;READ HEADER AND DATA=72
5363  026324  005037  001774                  CLR     @#ERFLG$         ;CLEAR ERROR FLAG
5364  026330  004737  045026                  JSR     PC,@#COMHD       ;READ HEADER AND DATA
5365
5366
5367                                          ;IF THE PROGRAM COMES BACK HERE WITHOUT ERROR PRINTOUTS
5368                                          ;FROM THE "COMHD" ROUTINE THAT MEANS SECTOR GAP,
5369                                          ;FIRST SYNC, HEADER, HEADER CRC, HEADER GAP AND
5370                                          ;SYNC BYTE HAVE GONE BY AND SYNCS WERE CORRECTLY
5371                                          ;DETECTED.
5372
5373
5374                                          ;RHWC IS CHECKED TO BE = 0 AFTER THE READ OPERATION
5375
5376  026334  017767  153264  152564          MOV     @RHWC,$BODAT     ;LOAD AND TEST RHWC
5377  026342  001401                          BEQ     20$              ;SHOULD = 0
5378  026344  104040                          ERROR   40               ;RHWC DOES NOT = 0 AFTER A READ
5379                                                                    ;HEADER AND DATA
5380
5381                                          ;HEADER AND DATA ARE TO BE CHECKED.
5382                                          ;IN CHECKING READ DATA THE WRITE FROM BUFFER
5383                                          ;"WRFROM" IS FILLED WITH EXPECTED DATA AND
5384                                          ;COMPARISONS ARE MADE.
5385
5386  026346  005737  001774          20$:    TST     @#ERFLG$         ;ANY ERRORS ALREADY THERE
5387  026352  001046                          BNE     TST64            ;BRANCH IF YES
5388  026354  004737  040746                  JSR     PC,@#CHECKE      ;CHECK THAT BITS = 1
5389  026360  104400  057176                  TYPE    ,CPHALT          ;CANNOT CONTINUE TESTING IF THEY DON'T
5390  026364  000000                          HALT                     ;STOP THE TEST
5391  026366  012700  002070                  MOV     #WRFROM,R0       ;GETTING READY TO FILL EXPECTED DATA
5392  026372  012720  010000                  MOV     #0!FMT22,(R0)+ ;CYLINDER 0
5393  026376  112746  000001                  MOVB    #1,-(SP)         ;IN LOWER BYTE GET SECTOR
5394  026402  112766  000001  000001          MOVB    #1,1(SP)         ;GET TRACK IN HIGHER BYTE
5395  026410  012620                          MOV     (SP)+,(R0)+      ;GET TRACK/SECTOR IN BUFFER
5396  026412  012720  000000                  MOV     #0,(R0)+         ;KEY1 IN BUFFER
5397  026416  012720  000000                  MOV     #0,(R0)+         ;KEY2 IN BUFFER
5398  026422  012701  000400                  MOV     #256.,R1         ;DATA WORD COUNTER
5399  026426  012702  052525                  MOV     #052525,R2               ;DATA
5400  026432  010220                  3$:     MOV     R2,(R0)+         ;DATA INTO BUFFER
5401  026434  005301                          DEC     R1               ;COUNT
5402  026436  001375                          BNE     3$               ;BRANCH IF 256 NOT DONE
5403
5404                                          ;NOW READ DATA BUFFER WILL BE CHECKED
```

```
5405
5406  026440  004037  041416                  JSR     R0,@#COMPAR     ;CHECK
5407  026444  002070                          WRFROM                  ;GOOD BUFFER
5408  026446  003134                          REINTO                  ;TEST BUFFER
5409  026450  000404                          4+256.                  ;NUMBER OF WORDS CHECKED
5410  026452  026460                          4$                      ;RETURN POINT FOR ERROR HEADER
5411  026454  026464                          5$                      ;RETURN POINT FOR ERROR DATA
5412  026456  026470                          TST64                   ;RETURN FOR GOOD COMPARISON
5413  026460  104004                  4$:     ERROR   4               ;READ NEXT ERROR
5414  026462  000207                          RTS     PC              ;RETURN TO "COMPAR"
5415  026464  104005                  5$:     ERROR   5               ;WORD NOS 1 TO 4 ARE
5416                                                                  ;HEADER WORDS
5417                                                                  ;5 TO 260 ARE DATA WORDS
5418  026466  000207                          RTS     PC              ;RETURN TO "COMPAR"
5419
5420
5421
5422
```

```
5423
5424                            ;*****************************************************************
5425                            ;*TEST 64        WRITE DATA
5426
5427                            ;*      WRITE CYLINDER 0, FORMAT 16 BITS PER WORD
5428                            ;*      TRACK 0, SECTOR 0, KEYS 0, NUMBER OF WORDS 256 OF 377
5429                            ;*      ANY ERROR LOGIC INDICATION IS NOT CONCLUSIVE ON FIRST PASS
5430                            ;*      BECAUSE ERROR LOGIC HAS NOT BEEN CHECKED
5431
5432                            ;*****************************************************************
5433  026470  000004            TST64:  SCOPE
5434
5435  026472  012737  000064  002020        MOV     #TTNO,@#TSTNM   ;THIS SAVES TEST NUMBER
5436
5437
5438  026500  012706  001000                MOV     #STACK,SP       ;RESET STACK
5439  026504  012737  000064  002020        MOV     #TTNO,@#TSTNM   ;THIS SAVES TEST NUMBER
5440
5441  026512  004037  040440                JSR     R0,@#CLAREA     ;CLEAR SIMULATED DISK
5442  026516  047070                        .WORD   DISK            ;FROM
5443  026520  050114                        .WORD   TOLGAP+16       ;TO
5444  026522  000000                        .WORD   0               ;DATA
5445
5446                                        ;THESE ARE SETUP FOR DISKLESS USE ONLY
5447
5448  026524  012737  010000  045152        MOV     #0!FMT22,@#CYL;CYLINDER 0
5449                                                                ;16 BITS PER WORD
5450  026532  112737  000000  045155        MOVB    #0,@#SECOTR+1;TRACK 0
5451  026540  112737  000000  045154        MOVB    #0,@#SECOTR     ;SECTOR 0
5452  026546  005037  045156                CLR     @#KEY1          ;KEY1 0
5453  026552  005037  045160                CLR     @#KEY2          ;KEY2 0
5454  026556  012737  000400  045220        MOV     #256.,@#NOWORD  ;NO OF DATA WORDS
5455  026564  012737  000001  045162        MOV     #1,@#X          ;WRITE DATA
5456  026572  004537  041730                JSR     R5,@#CRC        ;GO TO CALCULATE CRC
5457  026576  045152                        CYL
5458  026600  047052                        WCRC
5459
5460                                ;THESE ARE REGULAR SETUPS
5461
5462
5463  026602  004037  040440                JSR     R0,@#CLAREA     ;FILL WRITE BUFFER WITH 377
5464  026606  002070                        WRFROM                  ;FROM LOCATION
5465  026610  003070                        WRFROM+<256.*2>         ;TO LOCATION
5466  026612  000377                        377                     ;DATA
5467  026614  004737  040522                JSR     PC,@#CLDISK     ;SETUP GENERAL REGISTERS
5468  026620  012777  177400  152776        MOV     #-256.,@RHWC    ;256. DATA WORDS
5469  026626  012777  002070  152772        MOV     #WRFROM,@RHBA   ;STARTING ADDRESS OF WRITE BUFFER
5470  026634  012746  000000                MOV     #0,-(SP)        ;SECTOR 0
5471  026640  112766  000000  000001        MOVB    #0,1(SP)        ;TRACK 0
5472  026646  012677  152764                MOV     (SP)+,@RHDST    ;SECTOR 0 TRACK 0
5473  026652  012777  010000  152762        MOV     #FMT22,@RHOF    ;16 BITS PER WORD FORMAT
5474  026660  012777  000000  152756        MOV     #0,@RHCA        ;CYLINDER 0
5475  026666  004737  040556                JSR     PC,@#CHECKT     ;CHECK THAT DVA,RDY,DPR,DRY = 1
5476  026672  104400  057176                TYPE    ,CPHALT         ;AND THAT NO OTHERS = 1.  CANNOT CON-
```

```
                                                                ;TINUE TESTING IF BOTH AREN'T TRUE
026676  000000                          HALT                    ;STOP THE TEST
026700  013711  002044                  MOV     @#WRIDAT,@R1    ;WRITE DATA=60
026704  005037  001774                  CLR     @#ERFLGS        ;CLEAR ERROR FLAG
026710  004737  045026                  JSR     PC,@#COMMD      ;WRITE DATA


                        ;IF THE PROGRAM COMES BACK HERE WITHOUT ERROR PRINTOUTS
                        ;FROM THE "COMMD" 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.

026714  004737  040126                  JSR     PC,@#PUTREG     ;SAVE REGISTERS
026720  005737  001774                  TST     @#ERFLGS        ;HAVE ANY ERRORS OCCURED?
026724  001041                          BNE     TST65           ;BRANCH IF YES
026726  012700  000377                  MOV     #377,R0         ;GOOD DATA
026732  012701  047070                  MOV     #DISK,R1        ;DATA WRITTEN INTO "DISK"
026736  012702  000400                  MOV     #256.,R2        ;COUNTER

026742  012737  000401  045272  1$:     MOV     #256.+1,@#ERWORD;FOR ERROR WORD
026750  020021                          CMP     R0,(R1)+        ;COMPARE GOOD DATA WITH DATA ON DISK
026752  001424                          BEQ     3$              ;BRANCH IF GOOD
026754  010037  001124                  MOV     R0,@#GDDAT      ;GOOD DATA
026760  014137  001126                  MOV     -(R1),@#BDDAT   ;BAD DATA
026764  160237  045272                  SUB     R2,@#ERWORD     ;ERROR WORD NO
026770  005737  001774                  TST     @#ERFLGS        ;ANY ERRORS ALREADY THERE?
026774  001002                          BNE     2$              ;BRANCH IF YES
026776  104004                          ERROR   4               ;ERROR ON WRITE DATA COMMAND
027000  000401                          BR      64$             ;BRANCH TO AVOID PRINTING NEXT ERROR

027002  104005          2$:     ERROR   5               ;WORD NO GIVES WORD IN ERROR
027004  005721          64$:    TST     (R1)+           ;UNDO -(R1) FOR BAD DATA
027006  013746  177570          MOV     @#SWR,-(SP)     ;GET SWITCH SETTING
027012  042716  177177          BIC     #177177,(SP)    ;KEEP ONLY SWITCH 7 AND 8
027016  022726  000200          CMP     #SW07,(SP)+     ;IS 7 SET AND 8 RESET
027022  001402                  BEQ     TST65           ;BRANCH OUT IF YES
027024  005302          3$:     DEC     R2              ;IF NOT COUNT 256 WORDS
027026  001345                  BNE     1$              ;BRANCH IF 256. NOT DONE
```

```
                                        ;*****************************************************************
                                        ;*TEST 65       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)

                                        ;*****************************************************************
027030  000004                  TST65:  SCOPE
027032  012706  001000                  MOV     #STACK,SP       ;RESET STACK

027036  012737  000065  002020          MOV     #TTNO,@#TSTNM   ;THIS SAVES TEST NUMBER
027044  004037  040440                  JSR     R0,@#CLAREA     ;CLEAR SIMULATED DISK
027050  047070                          .WORD   DISK            ;FROM
027052  050066                          .WORD   DISK+776        ;TO
027054  177400                          .WORD   177400          ;DATA

027056  004037  040440                  JSR     R0,@#CLAREA     ;CLEAR READ INTO BUFFER
027062  003134                          .WORD   REINTO          ;FROM
027064  004132                          .WORD   REINTO+776      ;TO
027066  000000                          .WORD   0               ;DATA

                                        ;THESE ARE TO SETUP FOR DISKLESS USE ONLY

027070  012737  010000  045152          MOV     #FMT22,@#CYL    ;CYLINDER 0 16 BITS PER WORD FORMAT
027076  105037  045155                  CLRB    @#SECOTR+1      ;TRACK 0
027102  112737  000001  045154          MOVB    #1,@#SECOTR     ;SECTOR 1
027110  005037  045156                  CLR     @#KEY1          ;KEY1=0
027114  005037  045160                  CLR     @#KEY2          ;KEY2=0
027120  012737  000012  045232          MOV     #10.,@#DAWORD   ;NO. OF DATA WORDS
027126  005037  045162                  CLR     @#X             ;THIS IS A READ COMMAND
027132  004537  041730                  JSR     R5,@#CRC        ;GO TO CALCULATE CRC
027136  045152                          CYL
027140  047052                          WCRC

                                        ;THESE ARE REGULAR SETUPS

027142  004737  040522                  JSR     PC,@#CLDISK     ;SETUP GENERAL REGISTERS
027146  013711  002050                  MOV     @#READAT,@R1    ;READ DATA INTO RHCS1=70
027152  012777  177766  152444          MOV     #-10.,@RHWC     ;10 DATA WORDS
027160  012777  003134  152440          MOV     #REINTO,@RHBA   ;STARTING ADDRESS OF READ BUFFER
027166  112746  000001                  MOVB    #1,-(SP)        ;IN LOWER BYTE GET SECTOR 1
027172  112766  000000  000001          MOVB    #0,1(SP)        ;GET TRACK0 IN UPPER BYTE
027200  012677  152432                  MOV     (SP)+,@RHDST    ;TRACK/SECTOR IN RHDST
027204  012777  014000  152430          MOV     #FMT22!ECI,@RHOF ;16 BITS PER WORD
                                                                ;ECC CORRECTION INHIBIT BECAUSE
                                                                ;ECC IS NOT CHECKED HERE
027212  005077  152426                  CLR     @RHCA           ;CYLINDER 0
027216  004737  040556                  JSR     PC,@#CHECKT     ;CHECK THAT DVA,RDY,DPR,DRY = 1
027222  104400  057176                  TYPE    ,CPHALT         ;AND THAT NO OTHERS = 1.  CANNOT CON-
                                                                ;TINUE TESTING IF BOTH AREN'T TRUE
027226  000000                          HALT                    ;STOP THE TEST
```

```
5573  027230  005037  001774                  CLR     @#ERFLG$        ;CLEAR ERROR FLAG
5574  027234  004737  045026                  JSR     PC,@#COMMD      ;READ DATA
5575
5576                                          ;IF THE PROGRAM COMES BACK HERE WITHOUT ERROR PRINTOUT
5577                                          ;FROM "COMMD" ROUTINE IN MEANS DATA IS TO BE CHECKED
5578
5579                                          ;NOW THE DATA READ INTO "REINTO" BUFFER WILL
5580                                          ;BE CHECKED, ONLY 10 WORDS SHOULD BE CHANGED
5581                                          ;ALL OTHER WORDS SHOULD REMAIN UNCHANGED
5582                                          ;THE "WRFROM" BUFFER IS FILLED WITH EXPECTED DATA AND CHECKED
5583
5584  027240  005737  001774                  TST     @#ERFLG$        ;HAVE ANY ERRORS OCCURED?
5585  027244  001053                          BNE     TST66           ;;BRANCH IF YES
5586  027246  004037  040440                  JSR     R0,@#CLAREA     ;CLEAR BUFFER
5587  027252  002070                          WRFROM                  ;FROM
5588  027254  003066                          WRFROM+776              ;TO
5589  027256  000000                          0                       ;DATA
5590
5591  027260  004037  040440                  JSR     R0,@#CLAREA     ;FILL EXPECTED DATA
5592  027264  002070                          WRFROM                  ;FROM
5593  027266  002112                          WRFROM+22               ;TO
5594  027270  177400                          177400                  ;DATA
5595
5596                                          ;NOW READ DATA BUFFER IS CHECKED
5597
5598  027272  012700  002070                  MOV     #WRFROM,R0      ;GOOD DATA
5599  027276  012701  003134                  MOV     #REINTO,R1      ;DATA READ
5600  027302  012702  000400                  MOV     #256.,R2        ;COUNTER
5601  027306  012737  000401  045272  1$:     MOV     #257.,@#ERWORD  ;FOR ERROR WORD NO
5602  027314  022021                          CMP     (R0)+,(R1)+     ;COMPARE GOOD WITH READ BUFFER
5603  027316  001424                          BEQ     2$              ;BRANCH IF GOOD
5604  027320  014037  001124                  MOV     -(R0),@#$GDDAT  ;GOOD DATA
5605  027324  014137  001126                  MOV     -(R1),@#$BDDAT  ;BAD DATA
5606  027330  160237  045272                  SUB     R2,@#ERWORD     ;ERROR WORD NO
5607  027334  005737  001774                  TST     @#ERFLG$        ;ANY ERRORS ALREADY THERE
5608  027340  001002                          BNE     3$              ;IF YES BRANCH DO NOT TYPE HEADER
5609  027342  104004                          ERROR   4               ;ERROR ON READ DATA
5610  027344  000401                          BR      4$              ;BRANCH TO AVOID PRINTING NEXT ERROR
5611  027346  104005                  3$:     ERROR   5               ;WORD NO 1-10 ARE DATA
5612                                                                  ;WORDS
5613                                                                  ;WORD NOS 11-256 HAVE NOT BEEN
5614                                                                  ;READ AND BUFFER SHOULD BE
5615                                                                  ;ZERO IF OTHER THAN ZERO
5616                                                                  ;WRONG NUMBER OF WORDS HAVE
5617                                                                  ;BEEN READ IN THE DISK NOW
5618                                                                  ;CONTAINS 177400 ALL 256
5619                                                                  ;WORDS BUT ONLY 10 WORDS
5620                                                                  ;SHOULD BE READ IN
5621
5622  027350  022021                  4$:     CMP     (R0)+,(R1)+     ;UNDO -(R0) AND -(R1) FOR ERROR
5623  027352  013746  177570                  MOV     @#SWR,-(SP)     ;GET SWITCH SETTING
5624  027356  042716  177177                  BIC     #177177,(SP)    ;KEEP ONLY SWITCH 7 AND 8
5625  027362  022726  000200                  CMP     #SW07,(SP)+     ;IS 7 SET AND 8 RESET
5626  027366  001402                          BEQ     TST66           ;BRANCH OUT IF YES
```

```
5627  027370  005302                  2$:     DEC     R2              ;COUNT
5628  027372  001345                          BNE     1$              ;BRANCH IF NOT COMPLETE
5629
5630
5631
```

```
                                ;
                                ;*****************************************************************
                                ;*TEST 66       WRITE CHECK HEADER AND DATA
                                ;
                                ;*      WRITE CHECK CYLINDER 0, FORMAT 16 BITS PER WORD
                                ;*      TRACK 1, SECTOR 1, KEYS 0, 36 WORDS AS SHOWN BELOW
                                ;*      ANY DEVICE LOGIC ERROR INDICATION IS NOT CONCLUSIVE ON FIRST PASS
                                ;*      BECAUSE ERROR LOGIC HAS NOT BEEN CHECKED YET (ON FIRST PASS)
                                ;*      ONLY RH WRITE CHECK ERROR (RHCS2 BIT 14) IS TESTED HERE
                                ;*****************************************************************
027374  000004                  TST66:  SCOPE
                                ;       DATA TABLE
                                ;       20 WORDS OF 070707
                                ;       THEN 16 WORDS WITH ZERO FLOATING FROM RIGHT
                                ;       TO LEFT (EG. 177776,177775,177773 ETC)
027376  012706  001000                  MOV     #STACK,SP       ;RESET STACK
027402  012737  000066  002020          MOV     #TTNO,@#TSTNM   ;THIS SAVES TEST NUMBER
                                ;SET UP "REINTO" FOR WHAT IS TO BE READ
027410  012701  003134                  MOV     #REINTO,R1      ;STARTING ADDRESS
027414  012721  010000                  MOV     #FMT22,(R1)+    ;CYLINDER 0 FORMAT 16 BIT WORDS
027420  012721  000401                  MOV     #401,(R1)+      ;TRACK=1, SECTOR=1
027424  005021                          CLR     (R1)+           ;KEY1=0
027426  005021                          CLR     (R1)+           ;KEY2=0
027430  004037  040440                  JSR     R0,@#CLAREA     ;FILL "REINTO" BUFFER
027434  003144                          .WORD   REINTO+(4*2)    ;FROM
027436  003212                          .WORD   REINTO+(23.*2)  ;TO
027440  070707                          .WORD   070707          ;DATA
027442  012700  177776                  MOV     #177776,R0      ;GETTING READY TO FLOAT 0
027446  012701  003214                  MOV     #REINTO+(24.*2),R1;STARTING ADDRESS WHERE 177776 GOES
027452  010021                  1$:     MOV     R0,(R1)+        ;MOVE IN FLOATING 0
027454  000261                          SEC                     ;SET CARRY
027456  006100                          ROL     R0              ;GET 0 ONE BIT LEFT
027460  103774                          BCS     1$              ;BRANCH IF 16 NOT DONE
027462  004037  040440                  JSR     R0,@#CLAREA     ;FILL THE REST OF BUFFER WITH 0
027466  003254                          .WORD   REINTO+(40.*2)  ;FROM
027470  004132                          .WORD   REINTO+776      ;TO
027472  000000                          .WORD   0               ;DATA
                                ;SET UP SIMULATED DISK WITH WHAT IS TO BE READ
027474  004037  040440                  JSR     R0,@#CLAREA     ;FILL "DISK" BUFFER
027500  047070                          .WORD   DISK            ;FROM
027502  047136                          .WORD   DISK+(19.*2)    ;TO
027504  070707                          .WORD   070707          ;DATA
027506  012700  177776                  MOV     #177776,R0      ;GETTING READY TO FLOAT ZEROS
027512  012701  047140                  MOV     #DISK+(20.*2),R1;STARTING ADDRESS WHERE 177776 GOES
```

```
5686  027516  010021            2$:   MOV   R0,(R1)+        ;MOVE IN FLOATING 0
5687  027520  000261                  SEC                   ;SET CARRY
5688  027522  006100                  ROL   R0              ;GET 0 ONE BIT LEFT
5689  027524  103774                  BCS   2$              ;BRANCH IF 16 NOT DONE
5690
5691  027526  004037  040440          JSR   R0,@#CLAREA     ;FILL THE REST OF BUFFER WITH 177777
5692  027532  047200                  .WORD DISK+(36.*2)    ;FROM
5693  027534  050066                  .WORD DISK+776        ;TO
5694  027536  177777                  .WORD 177777          ;DATA
5695
5696  027540  004737  041246          JSR   PC,@#WRCHHD     ;WRITE CHECK HEADER AND DATA
5697                                                        ;CYLINDER 0, TRACK 1, SECTOR 1
5698
5699                                  ;IF THE PROGRAM COMES BACK HERE THEN WRITE CHECK
5700                                  ;HAS BEEN COMPLETED NOW WRITE CHECK ERROR BIT IS TO BE TESTED
5701
5702  027544  013746  001762          MOV   @#UNIT,-(SP)    ;GET UNIT NUMBER
5703  027550  052716  000100          BIS   #IR,(SP)        ;ONLY BIT 6 SHOULD BE SET
5704  027554  004737  040126          JSR   PC,@#PUTREG     ;SAVE REGISTERS
5705  027560  022637  001700          CMP   (SP)+,@#CS2     ;COMPARE RHCS2
5706  027564  001406                  BEQ   4$              ;BRANCH IF GOOD
5707  027566  032712  040000          BIT   #WCE,@R2        ;WRITE CHECK ERROR HIGH?
5708  027572  001402                  BEQ   3$              ;BRANCH IF ERROR NOT DUE TO "WCE"
5709  027574  104017                  ERROR 17              ;RHDB CONTAINS FAILING WORD
5710  027576  000401                  BR    4$              ;RHBA CONTAINS ADDRESS+2
5711                                                        ;OF THE WORD IN MEMORY FROM
5712                                                        ;THE DISK THAT DID NOT COMPARE
5713                                                        ;TRE AND SC WILL BE SET DUE TO
5714                                                        ;WCE
5715  027600  104017            3$:   ERROR 17              ;WCE CORRECTLY WAS NOT SET BUT SOME
5716                                                        ;BITS OTHER THAN IR
5717                                                        ;AND UNIT NO. WAS SET
5718
5719                                  ;NOW CHECK MEMORY TO SEE IF NOTHING GOT DESTROYED
5720                                  ;FILL "WRFROM" WITH WHAT SHOULD BE IN "REINTO" THEN CHECK
5721
5722  027602  012700  002070    4$:   MOV   #WRFROM,R0      ;STARTING ADDRESS
5723  027606  012720  010000          MOV   #FMT22,(R0)+    ;CYLINDER
5724  027612  012720  000401          MOV   #401,(R0)+      ;TRACK=1, SECTOR=1
5725  027616  005020                  CLR   (R0)+           ;KEY1=0
5726  027620  005020                  CLR   (R0)+           ;KEY2=0
5727
5728  027622  004037  040440          JSR   R0,@#CLAREA     ;FILL "WRFROM" BUFFER
5729  027626  002100                  .WORD WRFROM+(4*2)    ;FROM
5730  027630  002146                  .WORD WRFROM+(23.*2)  ;TO
5731  027632  070707                  .WORD 070707          ;DATA
5732
5733  027634  012700  177776          MOV   #177776,R0      ;GETTING READY TO FLOAT 0
5734  027640  012701  002150          MOV   #WRFROM+(24.*2),R1 ;STARTING ADDRESS WHERE 177776 GOES
5735  027644  010021            5$:   MOV   R0,(R1)+        ;MOVE IN FLOATING 0
5736  027646  000261                  SEC                   ;SET CARRY
5737  027650  006100                  ROL   R0              ;GET 0 ONE BIT LEFT
5738  027652  103774                  BCS   5$              ;BRANCH IF 16 NOT DONE
5739
```

```
5740  027654  004037  040440              JSR     R0,@#CLAREA     ;FILL THE REST OF BUFFER WITH 0
5741  027660  002210                      .WORD   WRFROM+<40.*2>  ;FROM
5742  027662  003066                      .WORD   WRFROM+776      ;TO
5743  027664  000000                      .WORD   0               ;DATA
5744
5745                                      ;NOW THE READ BUFFER WILL BE CHECKED
5746  027666  005037  001774              CLR     @#ERFLG$        ;CLEAR ERROR FLAG
5747
5748  027672  004037  041416              JSR     R0,@#COMPAR     ;CHECK
5749  027676  002070                      WRFROM                  ;GOOD BUFFER
5750  027700  003134                      REINTO                  ;TEST BUFFER
5751  027702  000400                      256.                    ;NUMBER OF WORDS CHECKED
5752  027704  027712                      6$                      ;RETURN POINT FOR ERROR HEADER
5753  027706  027716                      7$                      ;RETURN POINT FOR ERROR DATA
5754  027710  027724                      TST67                   ;RETURN FOR GOOD COMPARISON
5755  027712  104004          6$:         ERROR   4               ;READ NEXT ERROR 5
5756  027714  000207                      RTS     PC              ;RETURN TO COMPARISON SUBROUTINE
5757  027716  104005          7$:         ERROR   5               ;DATA IN REINTO BUFFER GOT
5758                                                              ;CHANGED AFTER A WRITE
5759                                                              ;CHECK HEADER AND DATA COMMAND
5760                                                              ;WORD NO CONTAINS THE WORD
5761                                                              ;NUMBER THAT GOT CHANGED
5762  027720  000207                      RTS     PC              ;RETURN TO COMPARISON SUBROUTINE
5763
5764  027722  000240          10$:        NOP                     ;ONLY A BRANCH POINT
```

```
                                ;*****************************************************************
                                ;*TEST 67       WRITE CHECK DATA

                                ;*      WRITE CHECK DATA CYLINDER 0 FORMAT 16 BITS PER WORD
                                ;*      TRACK 1, SECTOR 1, KEYS 0, 32 WORDS OF DATA
                                ;*      ANY DEVICE LOGIC ERROR INDICATIONS ARE NOT CONCLUSIVE ON FIRST PASS
                                ;*      BECAUSE ERROR LOGIC HAS NOT YET (ON FIRST PASS) BEEN CHECKED
                                ;*      ONLY RH WRITE CHECK ERROR IS TESTED

                                ;*****************************************************************
027724  000004                  TST67:  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)

027726  012706  001000                  MOV     #STACK,SP       ;RESET STACK
027732  012737  000067  002020          MOV     #TTNO,@#TSTNM   ;THIS SAVES TEST NUMBER

                                        ;SET UP "REINTO" FOR WHAT IS TO BE READ

027740  012700  000001                  MOV     #1,R0           ;GETTING READY TO FLOAT 1
027744  012701  003134                  MOV     #REINTO,R1      ;STARTING ADDRESS WHERE 1 GOES
027750  010021          1$:             MOV     R0,(R1)+        ;MOVE FLOATING 1
027752  006100                          ROL     R0              ;GET 1 ONE BIT LEFT
027754  103375                          BCC     1$              ;BRANCH IF 16 NOT DONE
027756  012700  177776                  MOV     #177776,R0      ;GETTING READY TO FLOAT 0
027762  012701  003174                  MOV     #REINTO+<16.*2>,R1 ;STARTING ADDRESS WHERE 177776 GOES
027766  010021          2$:             MOV     R0,(R1)+        ;MOVE IN FLOATING 0
027770  000261                          SEC                     ;SET CARRY
027772  006100                          ROL     R0              ;GET 0 ONE BIT LEFT
027774  103774                          BCS     2$              ;BRANCH IF 16 NOT DONE

027776  004037  040440                  JSR     R0,@#CLAREA     ;FILL REST OF BUFFER WITH 1
030002  003234                          .WORD   REINTO+<32.*2>  ;FROM
030004  004132                          .WORD   REINTO+776      ;TO
030006  000001                          .WORD   1               ;WITH DATA

                                        ;SET UP SIMULATED DISK WITH WHAT IS TO BE READ

030010  012700  000001                  MOV     #1,R0           ;GETTING READY TO FLOAT 1
030014  012701  047070                  MOV     #DISK,R1        ;STARTING ADDRESS WHERE 1 GOES
030020  010021          3$:             MOV     R0,(R1)+        ;MOVE FLOATING 1
030022  006100                          ROL     R0              ;GET 1 ONE BIT LEFT
030024  103375                          BCC     3$              ;BRANCH IF 16 NOT DONE

030026  012700  177776                  MOV     #177776,R0      ;GETTING READY TO FLOAT 0
030032  012701  047130                  MOV     #DISK+<16.*2>,R1 ;STARTING ADDRESS WHERE 177776 GOES
030036  010021          4$:             MOV     R0,(R1)+        ;MOVE FLOATING 0
030040  000261                          SEC                     ;SET CARRY
030042  006100                          ROL     R0              ;GET 0 ONE BIT LEFT
```

```
5819  030044  103774                          BCS     4$              ;BRANCH IF 16 NOT DONE
5820
5821  030046  004037  040440                  JSR     R0,@#CLAREA     ;FILL REST OF BUFFER WITH 0
5822  030052  047170                          .WORD   DISK+(32.*2)    ;FROM
5823  030054  050066                          .WORD   DISK+776        ;TO
5824  030056  000000                          .WORD   0               ;WITH DATA
5825
5826  030060  004737  041560                  JSR     PC,@#WRCHDA     ;WRITE CHECK DATA
5827                                                                  ;CYLINDER 0, TRACK 1, SECTOR 1
5828                                                                  ;KEYS 0, 32 WORDS.
5829
5830                                          ;IF THE PROGRAM COMES BACK HERE THEN WRITE CHECK
5831                                          ;HAS BEEN COMPLETED NOW WRITE CHECK ERROR BIT IS TESTED
5832
5833  030064  013746  001762                  MOV     @#UNIT,-(SP)    ;GET UNIT NUMBER
5834  030070  052716  000100                  BIS     #IR,(SP)        ;ONLY BIT 6 SHOULD BE SET
5835  030074  004737  040126                  JSR     PC,@#PUTREG     ;SAVE REGISTERS
5836  030100  022637  001700                  CMP     (SP)+,@#CS2     ;COMPARE RHCS2
5837  030104  001407                          BEQ     6$              ;BRANCH IF GOOD
5838  030106  032737  040000  001700          BIT     #WCE,@#CS2      ;WRITE CHECK ERROR HIGH?
5839  030114  001402                          BEQ     5$              ;BRANCH IF ERROR NOT DUE TO "WCE"
5840  030116  104017                          ERROR   17              ;RHDB CONTAINS FAILING WORD
5841  030120  000401                          BR      6$              ;RHBA CONTAINS ADDRESS+2
5842                                                                  ;OF THE WORD IN MEMORY FROM
5843                                                                  ;THE DISK THAT DID NOT COMPARE
5844                                                                  ;TRE AND SC WILL BE SET DUE TO WCE
5845  030122  104017                  5$:     ERROR   17              ;WCE WAS CORRECTLY NOT SET
5846                                                                  ;BUT SOME BITS OTHER THAN
5847                                                                  ;IR AND UNIT NO. WERE SET
5848
5849                                          ;NOW CHECK MEMORY TO SEE IF ANYTHING GOT DESTROYED
5850                                          ;FILL "WRFROM" WITH WHAT SHOULD BE IN REINTO THEN CHECK
5851
5852  030124  005037  001774          6$:     CLR     @#ERFLG$        ;CLEAR ERROR FLAG
5853  030130  012700  000001                  MOV     #1,R0           ;GETTING READY TO FLOAT 1
5854  030134  012701  002070                  MOV     #WRFROM,R1      ;START ADDRESS WHERE 1 GOES
5855  030140  010021                  7$:     MOV     R0,(R1)+        ;MOVE FLOATING 1
5856  030142  006100                          ROL     R0              ;GET 1 ONE BIT LEFT
5857  030144  103375                          BCC     7$              ;BRANCH IF 16 NOT DONE
5858
5859  030146  012700  177776                  MOV     #177776,R0      ;GETTING READY TO FLOAT 0
5860  030152  012701  002130                  MOV     #WRFROM+(16.*2),R1 ;STARTING ADDRESS WHERE 177776 GOES
5861  030156  010021                  10$:    MOV     R0,(R1)+        ;MOVE IN FLOATING 0
5862  030160  000261                          SEC                     ;SET CARRY
5863  030162  006100                          ROL     R0              ;GET 0 ONE BIT LEFT
5864  030164  103774                          BCS     10$             ;BRANCH IF CARRY SET
5865
5866  030166  004037  040440                  JSR     R0,@#CLAREA     ;FILL REST OF BUFFER WITH 1
5867  030172  002170                          .WORD   WRFROM+(32.*2)  ;FROM
5868  030174  003066                          .WORD   WRFROM+776      ;TO
5869  030176  000001                          .WORD   1               ;WITH DATA
5870
5871                                          ;NOW THE READ BUFFER WILL BE CHECKED
5872
```

```
5873  030200  004037  041416                  JSR     R0,@#COMPAR     ;CHECK
5874  030204  002070                          WRFROM                  ;GOOD BUFFER
5875  030206  003134                          REINTO                  ;TEST BUFFER
5876  030210  000400                          256.                    ;NUMBER OF WORDS CHECKED
5877  030212  030220                          11$                     ;RETURN POINT FOR ERROR HEADER
5878  030214  030224                          12$                     ;RETURN POINT FOR ERROR DATA
5879
5880  030216  030232                          TST70                   ;RETURN FOR GOOD COMPARISON
5881
5882  030220  104004                  11$:    ERROR   4               ;READ NEXT ERROR 5
5883  030222  000207                          RTS     PC              ;RETURN TO COMPARISON SUBROUTINE
5884  030224  104005                  12$:    ERROR   5               ;DATA IN REINTO BUFFER GOT
5885                                                                  ;CHANGED AFTER A WRITE
5886                                                                  ;CHECK DATA COMMAND
5887                                                                  ;WORD NO CONTAINS THE WORD
5888                                                                  ;NUMBER THAT GOT CHANGED
5889  030226  000207                          RTS     PC              ;RETURN TO COMPARISON SUBROUTINE
5890
5891  030230  000240                  13$:    NOP                     ;ONLY A BRANCH POINT
5892
5893
```

```
                                .SBTTL  ERROR BIT FUNCTIONAL TESTS



                                ;*****************************************************************
                                ;*TEST 70        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

                                ;*****************************************************************
030232  000004                TST70:  SCOPE

030234  012706  001000                MOV     #STACK,SP       ;RESET STACK
030240  012737  000070  002020        MOV     #TTNO,@#TSTNM   ;THIS SAVES TEST NUMBER
030246  004737  040522                JSR     PC,@#CLDISK     ;CLEAR DISK REGISTERS
030252  004737  040556                JSR     PC,@#CHECKT     ;CHECK THAT DVA,RDY,DPR,DRY = 1
030256  104400  057176                TYPE    ,CPHALT         ;AND THAT NO OTHERS = 1.  CANNOT CON-
                                                              ;TINUE TESTING IF BOTH AREN'T TRUE
030262  000000                        HALT                    ;STOP THE TEST



030264  012700  003134                MOV     #REINTO,R0      ;BUFFER STARTING FOR 3 ERROR
                                                              ;REGISTERS
030270  013720  001634                MOV     @#RHER1,(R0)+   ;RHER1 STORED IN REINTO
030274  012720  000000                MOV     #0,(R0)+        ;BITS NOT TO BE CHECKED IN RHER1
030300  013720  001640                MOV     @#RHER2,(R0)+   ;RHER2 STORED IN REINTO+4
030304  012720  000000                MOV     #0,(R0)+        ;BITS NOT TO BE CHECKED IN RHER2
030310  013720  001646                MOV     @#RHER3,(R0)+   ;RHER3 STORED IN REINTO+10
030314  012720  000000                MOV     #0,(R0)+        ;BITS NOT TO BE CHECKED IN RHER3

030320  013704  001650                MOV     @#RHAS,R4       ;R4 HAS RHAS
030324  013705  002004                MOV     @#ATTENT,R5     ;R5 HAS ATA BIT IN RHAS
030330  012737  030356  001110        MOV     #2$,@#SLPERR    ;THAT SHOULD SET WITH ERROR
                                                              ;RETURN POINT TO ERROR
030336  012737  000003  001172        MOV     #3,@#STMP1      ;ERROR REGISTER COUNTER
030344  012700  003134                MOV     #REINTO,R0      ;REGISTER BUFFER POINTER

030350  012002                1$:     MOV     (R0)+,R2        ;R2 HAS ADDRESS OF ERROR REG
030352  012701  000001                MOV     #BIT0,R1        ;R1 WILL HAVE BIT UNDER TEST
030356  052777  000040  151244  2$:   BIS     #CLR,@RHCS2     ;CLEAR RHCS2
030364  013777  001762  151236        MOV     @#UNIT,@RHCS2   ;REINSTATE UNIT NO.
030372  010112                        MOV     R1,@R2          ;SET ERROR BIT
030374  004737  040126                JSR     PC,@#PUTREG     ;READ AND SAVE REGISTERS
```

```
5948  030400  120537  001720                    CMPB    R5,@#AS          ;ONLY THE BIT IN R5 SHOULD BE
5949                                                                       ;SET IN RHAS
5950  030404  001401                            BEQ     3$               ;LOOK @ RHDS1 IF GOOD
5951  030406  104020                            ERROR   20               ;WITH THE SETTING OF ONE
5952                                                                       ;ERROR BIT IN AN ERROR
5953                                                                       ;REGISTER, THE CORRESPONDING
5954                                                                       ;RHAS BIT DID NOT SET
5955
5956  030410  013746  001724          3$:     MOV     @#DS1,-(SP)      ;GET RHDS1
5957  030414  042716  001100                  BIC     #VV!PROG,(SP)    ;REMOVE VV AND PROG
5958  030420  022726  140600                  CMP     #ATA!ERR!DPR!DRY,(SP)+;THESE BITS PLUS VV SHOULD BE IN RHDS1
5959  030424  001401                          BEQ     4$               ;CHECK 'GO' NEXT, IF THIS WAS OK
5960  030426  104020                          ERROR   20               ;WITH THE SETTING OF ONE
5961                                                                       ;ERROR BIT, COMPOSITE ERROR
5962                                                                       ;OR ATTENTION ACTIVE, OR
5963                                                                       ;ONE OF THE OTHER
5964                                                                       ;PERMANENT BITS DID NOT SET
5965
5966  030430  012777  000001  151174  4$:     MOV     #GO,@RHCS1       ;GIVE NO-OP
5967  030436  004737  040126                  JSR     PC,@#PUTREG      ;SAVE REGISTERS
5968  030442  020112                          CMP     R1,@R2           ;GO SHOULD NOT CLEAR ERROR
5969  030444  001410                          BEQ     5$               ;FURTHER CHECK OF 'GO' FUNCTIONALITY
5970  030446  010237  040172                  MOV     R2,@#REGADR      ;FAILING REGISTER
5971  030452  010137  001124                  MOV     R1,@#$GDDAT      ;GOOD DATA
5972  030456  013737  001700  001126          MOV     @#CS2,@#$BDDAT   ;BAD DATA
5973  030464  104001                          ERROR   1                ;"GO" WITH NO-OP CHANGED
5974                                                                       ;ERROR REGISTER
5975
5976  030466  013746  001724          5$:     MOV     @#DS1,-(SP)      ;GET RHDS1
5977  030472  042716  001100                  BIC     #VV!PROG,(SP)    ;CLEAR VV AND PROG
5978  030476  022726  140600                  CMP     #ATA!ERR!DPR!DRY,(SP)+;GO SHOULD NOT CLEAR ANY BITS
5979  030502  001401                          BEQ     7$               ;CHECK NEXT ERROR BIT IF A-OK
5980  030504  104020                          ERROR   20               ;"GO" WITH NO-OP SHOULD NOT CLEAR
5981                                                                       ;ATA AND/OR ERR
5982
5983
5984                                  ;THIS IS THE MAIN BIT TESTING CONTROL LOGIC
5985
5986
5987  030506  006301                  7$:     ASL     R1               ;GET NEXT BIT TO THE LEFT
5988  030510  103403                          BCS     10$              ;GO ON TO NEXT REGISTER IF DONE
5989  030512  031001                          BIT     (R0),R1          ;IS THIS BIT TO BE TESTED ?
5990  030514  001374                          BNE     7$               ;IF NOT, GET NEXT ONE
5991  030516  000717                          BR      2$               ;IF TO BE TESTED, GO DO IT !
5992
5993  030520  005720                  10$:    TST     (R0)+            ;ADVANCE R0 TO NEXT ERROR REG.
5994  030522  005337  001172                  DEC     @#STMP1          ;REGISTER COUNTER
5995  030526  001310                          BNE     1$               ;DO NEXT ONE, IF 3 NOT COMPLETE
5996
5997
5998                                          ;NOW AFTER SETTING ATA IN RHDS1 "1" IN RHAS AT THE
5999                                          ;DRIVE POSITION SHOULD CLEAR ATA IN RHDS1
6000
6001  030530  004737  040522          11$:    JSR     PC,@#CLDISK      ;CLEAR
```

```
6002  030534  012767  030530  150346          MOV     #11$,$LPERR     ;ERROR RETURN
6003  030542  012714  177777                  MOV     #-1,@R4         ;SET BIT IN RHAS AND ATA IN RHDS1
6004  030546  013777  002004  151074          MOV     @#ATTENT,@RHAS  ;WRITE 1 INTO  DRIVE BIT POSITION
6005  030554  004737  040126                  JSR     PC,@#PUTREG     ;SAVE REGISTERS
6006  030560  105737  001720                  TSTB    @#AS            ;THIS SHOULD BE ZERO
6007  030564  001401                          BEQ     12$
6008  030566  104020                          ERROR   20              ;MOVING A "1" INTO RHAS
6009                                                                  ;AT THE DRIVE BIT POSITION
6010                                                                  ;DID NOT CLEAR IT
6011
6012  030570  013746  001724          12$:    MOV     @#DS1,-(SP)     ;GET RHDS1
6013  030574  042716  001000                  BIC     #PROG,(SP)      ;MASK PROGRAMABLE
6014  030600  022726  040700                  CMP     #ERR!VV!DPR!DRY,(SP)+
6015                                                                  ;RHDS1 SHOULD HAVE THESE BITS
6016                                                                  ;BUT ATA SHOULD BE CLEARED
6017  030604  001401                          BEQ     13$             ;CHECK RHER1 IF GOOD
6018  030606  104020                          ERROR   20              ;MOVING "1" INTO RHAS AT THE
6019                                                                  ;DRIVE BIT POSITION DID NOT
6020                                                                  ;CLEAR ATA IN RHDS1
6021
6022  030610  022737  177777  001704  13$:    CMP     #-1,@#ER1       ;RHER1 SHOULD NOT CHANGE
6023                                                                  ;BY CLEARING RHAS
6024  030616  001401                          BEQ     TST71           ;BRANCH IF GOOD
6025  030620  104020                          ERROR   20              ;RHER1 WAS CHANGED BY CLEARING
6026                                                                  ;RHAS BY MOVING "1" INTO
6027                                                                  ;THE DRIVE BIT POSITION
6028
6029
6030
6031
6032
```

```
                                ;*****************************************************************
                                ;*TEST 71       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)
                                ;
                                ;*****************************************************************
030622  000004                TST71:  SCOPE
030624  012706  001000                MOV     #STACK,SP       ;RESET STACK
030630  012737  000071  002020        MOV     #TTNO,@#TSTNM   ;THIS SAVES TEST NUMBER
030636  004037  040440                JSR     R0,@#CLAREA     ;CLEAR SIMULATED DISK
030642  047070                        .WORD   DISK            ;FROM
030644  050066                        .WORD   DISK+776        ;TO
030646  177400                        .WORD   177400          ;DATA
030650  004037  040440                JSR     R0,@#CLAREA     ;CLEAR READ INTO BUFFER
030654  003134                        .WORD   REINTO          ;FROM
030656  004132                        .WORD   REINTO+776      ;TO
030660  000000                        .WORD   0               ;DATA
                                ;THESE ARE TO SETUP FOR DISKLESS USE ONLY
030662  012737  010000  045152        MOV     #FMT22,@#CYL    ;CYLINDER 0 16 BITS PER WORD FORMAT
030670  105037  045155                CLRB    @#SECTOR+1      ;TRACK 0
030674  112737  000001  045154        MOVB    #1,@#SECTOR     ;SECTOR 1
030702  005037  045156                CLR     @#KEY1          ;KEY1=0
030706  005037  045160                CLR     @#KEY2          ;KEY2=0
030712  012737  000012  045232        MOV     #10.,@#DAWORD   ;NO. OF DATA WORDS
030720  005037  045162                CLR     @#X             ;THIS IS A READ COMMAND
030724  004537  041730                JSR     R5,@#CRC        ;GO TO CALCULATE CRC
030730  045152                        CYL
030732  047052                        WCRC
                                ;THESE ARE REGULAR SETUPS
030734  004737  040522                JSR     PC,@#CLDISK     ;SETUP GENERAL REGISTERS
030740  013711  002050                MOV     @#READAT,@R1    ;READ DATA INTO RHCS1=70
030744  012777  177766  150652        MOV     #-10.,@RHWC     ;10 DATA WORDS
030752  012777  003134  150646        MOV     #REINTO,@RHBA   ;STARTING ADDRESS OF READ BUFFER
030760  112746  000001                MOVB    #1,-(SP)        ;IN LOWER BYTE GET SECTOR 1
030764  112766  000000  000001        MOVB    #0,1(SP)        ;GET TRACK0 IN UPPER BYTE
030772  012677  150640                MOV     (SP)+,@RHDST    ;TRACK/SECTOR IN RHDST
030776  012777  014000  150636        MOV     #FMT22!ECI,@RHOF ;16 BITS PER WORD
                                                              ;ECC CORRECTION INHIBIT BECAUSE
                                                              ;ECC IS NOT CHECKED HERE
031004  005077  150634                CLR     @RHCA           ;CYLINDER 0
031010  004737  040556                JSR     PC,@#CHECKT     ;CHECK THAT DVA,RDY,DPR,DRY = 1
031014  104400  057176                TYPE    ,CPHALT         ;AND THAT NO OTHERS = 1.  CANNOT CON-
                                                              ;TINUE TESTING IF BOTH AREN'T TRUE
```

```
6087  031020  000000                          HALT                    ;STOP THE TEST
6088  031022  052777  000010  150600          BIS     #BAI,@RHCS2     ;SET BUS ADDRESS INHIBIT
6089  031030  005037  001774                  CLR     @#ERFLGS        ;CLEAR ERROR FLAG
6090  031034  004737  045026                  JSR     PC,@#COMMD      ;READ DATA
6091
6092                                          ;IF THE PROGRAM COMES BACK HERE WITHOUT ERROR PRINTOUT
6093                                          ;FROM "COMMD" ROUTINE IN MEANS DATA IS TO BE CHECKED
6094
6095                                          ;NOW THE DATA READ INTO "REINTO" BUFFER WILL
6096                                          ;BE CHECKED, ONLY ONE WORD SHOULD BE CHANGED
6097                                          ;ALL OTHER WORDS SHOULD REMAIN UNCHANGED
6098                                          ;THE "WRFROM" BUFFER IS FILLED WITH EXPECTED DATA AND CHECKED
6099
6100  031040  005037  001774                  CLR     @#ERFLGS        ;CLEAR FLAG
6101  031044  004037  040440                  JSR     R0,@#CLAREA     ;CLEAR BUFFER
6102  031050  002070                          WRFROM                  ;FROM
6103  031052  003066                          WRFROM+776              ;TO
6104  031054  000000                          0                       ;DATA
6105
6106                                          ;EXPECTED DATA IS 177400 IN FIRST LOCATION ONLY
6107  031056  012737  177400  002070          MOV     #177400,@#WRFROM        ;EXPECTED DATA
6108
6109                                          ;NOW READ DATA BUFFER IS CHECKED
6110
6111  031064  012700  002070                  MOV     #WRFROM,R0      ;GOOD DATA
6112  031070  012701  003134                  MOV     #REINTO,R1      ;DATA READ
6113  031074  012702  000400                  MOV     #256.,R2        ;COUNTER
6114  031100  012737  000401  045272  1$:     MOV     #257.,@#ERWORD  ;FOR ERROR WORD NO
6115  031106  022021                          CMP     (R0)+,(R1)+     ;COMPARE GOOD WITH READ BUFFER
6116  031110  001424                          BEQ     2$              ;BRANCH IF GOOD
6117  031112  014037  001124                  MOV     -(R0),@#$GDDAT  ;GOOD DATA
6118  031116  014137  001126                  MOV     -(R1),@#$BDDAT  ;BAD DATA
6119  031122  160237  045272                  SUB     R2,@#ERWORD     ;ERROR WORD NO
6120  031126  005737  001774                  TST     @#ERFLGS                ;ANY ERRORS ALREADY THERE
6121  031132  001002                          BNE     3$              ;IF YES BRANCH DO NOT TYPE HEADER
6122  031134  104004                          ERROR   4               ;ERROR ON READ DATA
6123  031136  000401                          BR      4$              ;BRANCH TO AVOID PRINTING NEXT ERROR
6124  031140  104005                  3$:     ERROR   5               ;WOI  NO 1-10 ARE DATA
6125                                                                  ;WORDS
6126                                                                  ;WORD NOS 11-256 HAVE NOT BEEN
6127                                                                  ;READ AND BUFFER SHOULD BE
6128                                                                  ;ZERO IF OTHER THAN ZERO
6129                                                                  ;WRONG NUMBER OF WORDS HAVE
6130                                                                  ;BEEN READ IN THE DISK NOW
6131                                                                  ;CONTAINS 177400 ALL 256
6132                                                                  ;WORDS BUT ONLY 10 WORDS
6133                                                                  ;SHOULD BE READ IN
6134
6135  031142  022021                  4$:     CMP     (R0)+,(R1)+     ;UNDO -(R0) AND -(R1) FOR ERROR
6136  031144  013746  177570                  MOV     @#SWR,-(SP)     ;GET SWITCH SETTING
6137  031150  042716  177177                  BIC     #177177,(SP)    ;KEEP ONLY SWITCH 7 AND 8
6138  031154  022726  000200                  CMP     #SW07,(SP)+     ;IS 7 SET AND 8 RESET
6139  031160  001402                          BEQ     TST72           ;BRANCH OUT IF YES
6140  031162  005302                  2$:     DEC     R2              ;COUNT
```

```
 6141  031164  001345                          BNE     1$              ;BRANCH IF NOT COMPLETE
 6142
 6143
 6144
 6145
 6146
```

```
                                        ;*****************************************************************
                                        ;*TEST 72       RHCS2 - BIT # 11 - NEM
                                        ;
                                        ;*      READ CYLINDER0, FORMAT 16 BITS PER WORD
                                        ;*      TRACK0, SECTOR 1, KEYS 0, 1 WORD OF 177400
                                        ;*      THIS IS DONE WITH BUS ADDRESS INHIBIT SET
                                        ;*      BUS ADDRESS USED IS 760000 THIS IS ALWAYS NON EXISTANT
                                        ;*      THIS SHOULD SET NEM
                                        ;
                                        ;*****************************************************************
031166  000004                          TST72:  SCOPE
031170  012706  001000                          MOV     #STACK,SP       ;RESET STACK
031174  012737  000072  002020                  MOV     #TTNO,@#TSTNM   ;THIS SAVES TEST NUMBER
031202  004037  040440                          JSR     R0,@#CLAREA     ;CLEAR SIMULATED DISK
031206  047070                                  .WORD   DISK            ;FROM
031210  050066                                  .WORD   DISK+776        ;TO
031212  177400                                  .WORD   177400          ;DATA
                                        ;THESE ARE TO SETUP FOR DISKLESS USE ONLY
031214  012737  010000  045152                  MOV     #FMT22,@#CYL    ;CYLINDER 0 16 BITS PER WORD FORMAT
031222  105037  045155                          CLRB    @#SECOTR+1      ;TRACK 0
031226  112737  000001  045154                  MOVB    #1,@#SECOTR     ;SECTOR 1
031234  005037  045156                          CLR     @#KEY1          ;KEY1=0
031240  005037  045160                          CLR     @#KEY2          ;KEY2=0
031244  012737  000001  045232                  MOV     #1.,@#DAWORD    ;NO. OF DATA WORDS
031252  005037  045162                          CLR     @#X             ;THIS IS A READ COMMAND
031256  004537  041730                          JSR     R5,@#CRC        ;GO TO CALCULATE CRC
031262  045152                                  CYL
031264  047052                                  WCRC
                                        ;THESE ARE REGULAR SETUPS
031266  004737  040522                          JSR     PC,@#CLDISK     ;SETUP GENERAL REGISTERS
031272  013711  002050                          MOV     @#READAT,@R1    ;READ DATA INTO RHCS1=70
031276  012777  177777  150320                  MOV     #-1.,@RHWC      ;10 DATA WORDS
031304  012777  160000  150314                  MOV     #160000,@RHBA   ;STARTING ADDRESS OF READ BUFFER
031312  052711  001400                          BIS     #A16!A17,@R1    ;IS 760000
031316  112746  000001                          MOVB    #1,-(SP)        ;IN LOWER BYTE GET SECTOR 1
031322  112766  000000  000001                  MOVB    #0,1(SP)        ;GET TRACK0 IN UPPER BYTE
031330  012677  150302                          MOV     (SP)+,@RHDST    ;TRACK/SECTOR IN RHDST
031334  012777  014000  150300                  MOV     #FMT22!ECI,@RHOF ;16 BITS PER WORD
                                                                        ;ECC CORRECTION INHIBIT BECAUSE
                                                                        ;ECC IS NOT CHECKED HERE
031342  005077  150276                          CLR     @RHCA           ;CYLINDER 0
031346  004737  040556                          JSR     PC,@#CHECKT     ;CHECK THAT DVA,RDY,DPR,DRY = 1
031352  104400  057176                          TYPE    ,CPHALT         ;AND THAT NO OTHERS = 1.  CANNOT CON-
                                                                        ;TINUE TESTING IF BOTH AREN'T TRUE
031356  000000                                  HALT                    ;STOP THE TEST
031360  052777  000010  150242                  BIS     #BAI,@RHCS2     ;SET BUS ADDRESS INHIBIT
031366  005037  001774                          CLR     @#ERFLG$        ;CLEAR ERROR FLAG
```

```
6201  031372  004737  045026              JSR     PC,@#COMND      ;READ DATA
6202
6203
6204
6205  031376  011137  001126      1$:     MOV     @R1,@#$BDDAT    ;TEST DATA
6206
6207  031402  022737  145670  001126      CMP     #SC!TRE!DVA!A16!A17!RDY!70,@#$BDDAT ;COMPARE RHCS1
6208  031410  001406                      BEQ     2$              ;BRANCH IF GOOD
6209  031412  012737  144270  001124      MOV     #SC!TRE!DVA!RDY!70,@#$GDDAT ;GOOD DATA
6210  031420  010137  040172              MOV     R1,@#REGADR     ;REGISTER RHCS1
6211  031424  104001                      ERROR   1               ;REFERENCE NON EXISTANT
6212                                              ;MEMORY DID NOT SET
6213                                              ;REQUIRED BITS
6214  031426  013746  001762      2$:     MOV     @#UNIT,-(SP)    ;GET UNIT NUMBER
6215  031432  052716  004110              BIS     #NEM!IR!BAI,(SP)        ;INCLUDE NEM BAI AND IR
6216  031436  012637  001124              MOV     (SP)+,@#$GDDAT  ;
6217  031442  011237  001126              MOV     @R2,@#$BDDAT    ;TEST DATA
6218  031446  023737  001124  001126      CMP     @#$GDDAT,@#$BDDAT;COMPARE RHCS2
6219  031454  001403                      BEQ     3$
6220  031456  010237  040172              MOV     R2,@#REGADR     ;REGISTER ADDRESS
6221  031462  104001                      ERROR   1               ;REFRENCING NONEXISTANT MEMORY
6222                                              ;CAUSED AN ERROR SHOULD SET NEM
6223  031464  017737  150136  001126  3$:     MOV     @RHBA,@#$BDDAT  ;TEST DATA
6224
6225  031472  022737  160000  001126      CMP     #160000,@#$BDDAT        ;COMPARE RHBA
6226  031500  001407                      BEQ     4$              ;BRANCH IF GOOD
6227  031502  012737  160000  001124      MOV     #160000,@#$GDDAT;GOOD DATA
6228  031510  013737  001626  040172      MOV     @#RHBA,@#REGADR ;REGISTER ADDRESS RHBA
6229  031516  104001                      ERROR   1               ;AFTER A NON EXISTANT MEMORY ERROR
6230                                              ;RHBA DOES NOT HAVE 160002
6231  031520                      4$:
6232
6233
6234
6235
6236
6237
6238
```

```
                                ;*****************************************************************
                                ;*TEST 73       WRITE CHECK ERROR

                                ;*      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

                                ;*****************************************************************
031520  000004                  TST73:  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)

031522  012706  001000                  MOV     #STACK,SP       ;RESET STACK

031526  012737  000073  002020          MOV     #TTNO,@#TSTNM   ;THIS SAVES TEST NUMBER
031534  004737  040522                  JSR     PC,@#CLDISK     ;INIT AND SET GENERAL REGISTERS
                                        ;SET UP "REINTO" FOR WHAT IS TO BE READ

031540  012700  000001                  MOV     #1,R0           ;GETTING READY TO FLOAT 1
031544  012701  003134                  MOV     #REINTO,R1      ;STARTING ADDRESS WHERE 1 GOES
031550  010021          1$:             MOV     R0,(R1)+        ;MOVE FLOATING 1
031552  006100                          ROL     R0              ;GET 1 ONE BIT LEFT
031554  103375                          BCC     1$              ;BRANCH IF 16 NOT DONE
031556  012700  177776                  MOV     #177776,R0      ;GETTING READY TO FLOAT 0
031562  012701  003174                  MOV     #REINTO+<16.*2>,R1 ;STARTING ADDRESS WHERE 177776 GOES
031566  010021          2$:             MOV     R0,(R1)+        ;MOVE IN FLOATING 0
031570  000261                          SEC                     ;SET CARRY
031572  006100                          ROL     R0              ;GET 0 ONE BIT LEFT
031574  103774                          BCS     2$              ;BRANCH IF 16 NOT DONE

031576  004037  040440                  JSR     R0,@#CLAREA     ;FILL REST OF BUFFER WITH 1
031602  003234                          .WORD   REINTO+<32.*2>  ;FROM
031604  004132                          .WORD   REINTO+776      ;TO
031606  000001                          .WORD   1               ;WITH DATA

                                        ;SET UP SIMULATED DISK WITH WHAT IS TO BE READ

031610  012700  000001                  MOV     #1,R0           ;GETTING READY TO FLOAT 1
031614  012701  047070                  MOV     #DISK,R1        ;STARTING ADDRESS WHERE 1 GOES
031620  010021          3$:             MOV     R0,(R1)+        ;MOVE FLOATING 1
031622  006100                          ROL     R0              ;GET 1 ONE BIT LEFT
031624  103375                          BCC     3$              ;BRANCH IF 16 NOT DONE

031626  012700  177776                  MOV     #177776,R0      ;GETTING READY TO FLOAT 0
031632  012701  047130                  MOV     #DISK+<16.*2>,R1 ;STARTING ADDRESS WHERE 177776 GOES
```

```
6293  031636  010021                    4$:   MOV   R0,(R1)+        ;MOVE FLOATING 0
6294  031640  000261                          SEC                   ;SET CARRY
6295  031642  006100                          ROL   R0              ;GET 0 ONE BIT LEFT
6296  031644  103774                          BCS   4$              ;BRANCH IF 16 NOT DONE
6297
6298  031646  004037  040440                JSR   R0,@#CLAREA     ;FILL REST OF BUFFER WITH 0
6299  031652  047170                          .WORD DISK+(32.*2)    ;FROM
6300  031654  050066                          .WORD DISK+776        ;TO
6301  031656  000000                          .WORD 0               ;WITH DATA
6302
6303                                          ;CHANGE FIFTH WORD TO 0 ON DISK
6304  031660  005037  047100                CLR   @#DISK+10       ;CLEAR FIFTH WORD ON DISK
6305  031664  005037  001774                CLR   @#ERFLG$        ;CLEAR ERROR FLAG
6306  031670  004737  041560                JSR   PC,@#WRCHDA     ;WRITE CHECK DATA
6307                                                                ;CYLINDER 0, TRACK 1, SECTOR 1
6308                                                                ;KEYS 0, 32 WORDS.
6309
6310                                          ;IF THE PROGRAM COMES BACK HERE THEN WRITE CHECK
6311                                          ;HAS BEEN COMPLETED NOW WRITE CHECK ERROR BIT IS TESTED
6312
6313  031674  013746  001762                MOV   @#UNIT,-(SP)    ;GET UNIT NUMBER
6314  031700  052716  040300                BIS   #IR!OR!WCE,(SP) ;ONLY BIT 6 SHOULD BE SET
6315  031704  004737  040126                JSR   PC,@#PUTREG     ;SAVE REGISTERS
6316  031710  022637  001700                CMP   (SP)+,@#CS2     ;COMPARE RHCS2
6317  031714  001407                          BEQ   6$              ;BRANCH IF GOOD
6318  031716  032737  040000  001700        BIT   #WCE,@#CS2      ;WRITE CHECK ERROR HIGH?
6319  031724  001002                          BNE   5$              ;BRANCH IF ERROR NOT DUE TO "WCE"
6320  031726  104017                          ERROR 17              ;RHDB CONTAINS FAILING WORD
6321  031730  000401                          BR    6$              ;RHBA CONTAINS ADDRESS+2
6322                                                                ;OF THE WORD IN MEMORY FROM
6323                                                                ;THE DISK THAT DID NOT COMPARE
6324                                                                ;TRE AND SC WILL BE SET DUE TO WCE
6325  031732  104017                    5$:   ERROR 17              ;WCE WAS CORRECTLY NOT SET
6326                                                                ;BUT SOME BITS OTHER THAN
6327                                                                ;IR AND UNIT NO. WERE SET
6328
6329  031734  022737  177745  001674  6$:   CMP   #-27.,@#WC       ;COMPARE RH#C AFTER A FORCED
6330                                                                ;WRITE CHEECK ERROR
6331  031742  001402                          BEQ   14$             ;BRANCH IF GOOD
6332  031744  104017                          ERROR 17              ;WORD COUNT REGISTER IN ERROR AFTER A
6333                                                                ;FORCED WRITECHECK ERROR ON FIFTH WORD
6334  031746  000405                          BR    15$             ;BRANCH TO CONTINUE
6335  031750  022737  003146  001676  14$:   CMP   #REINTO+(5*2),@#BA;COMPARE RHBA AFTER FORCED
6336                                                                ;WRITECHECK ERROR IN FIFTH WORD
6337  031756  001401                          BEQ   15$             ;BRANCH IF GOOD
6338  031760  104017                          ERROR 17              ;BUS ADDRESS REGISTER IN ERROR AFTER
6339                                                                ;FORCED WRITE CHECK ERROR ON FIFTH WORD
6340                                          ;NOW CHECK MEMORY TO SEE IF ANYTHING GOT DESTROYED
6341                                          ;FILL "WRFROM" WITH WHAT SHOULD BE IN REINTO THEN CHECK
6342
6343  031762  005037  001774          15$:   CLR   @#ERFLG$        ;CLEAR ERROR FLAG
6344  031766  012700  000001                MOV   #1,R0           ;GETTING READY TO FLOAT 1
6345  031772  012701  002070                MOV   #WRFROM,R1      ;START ADDRESS WHERE 1 GOES
6346  031776  010021                    7$:   MOV   R0,(R1)+        ;MOVE FLOATING 1
```

```
6347  032000  006100                          ROL     R0              ;GET 1 ONE BIT LEFT
6348  032002  103375                          BCC     7$              ;BRANCH IF 16 NOT DONE
6349
6350  032004  012700  177776                  MOV     #177776,R0      ;GETTING READY TO FLOAT 0
6351  032010  012701  002130                  MOV     #WRFROM+<16.*2>,R1 ;STARTING ADDRESS WHERE 177776 GOES
6352  032014  010021                  10$:    MOV     R0,(R1)+        ;MOVE IN FLOATING 0
6353  032016  000261                          SEC                     ;SET CARRY
6354  032020  006100                          ROL     R0              ;GET 0 ONE BIT LEFT
6355  032022  103774                          BCS     10$             ;BRANCH IF CARRY SET
6356
6357  032024  004037  040440                  JSR     R0,@#CLAREA     ;FILL REST OF BUFFER WITH 1
6358  032030  002170                          .WORD   WRFROM+<32.*2>  ;FROM
6359  032032  003066                          .WORD   WRFROM+776      ;TO
6360  032034  000001                          .WORD   1               ;WITH DATA
6361
6362                                          ;NOW THE READ BUFFER WILL BE CHECKED
6363
6364  032036  004037  041416                  JSR     R0,@#COMPAR     ;CHECK
6365  032042  002070                          WRFROM                  ;GOOD BUFFER
6366  032044  003134                          REINTO                  ;TEST BUFFER
6367  032046  000400                          256.                    ;NUMBER OF WORDS CHECKED
6368  032050  032056                          11$                     ;RETURN POINT FOR ERROR HEADER
6369  032052  032062                          12$                     ;RETURN POINT FOR ERROR DATA
6370
6371  032054  032070                          TST74                   ;RETURN FOR GOOD COMPARISON
6372
6373  032056  104004                  11$:    ERROR   4               ;READ NEXT ERROR 5
6374  032060  000207                          RTS     PC              ;RETURN TO COMPARISON SUBROUTINE
6375  032062  104005                  12$:    ERROR   5               ;DATA IN REINTO BUFFER GOT
6376                                          ;CHANGED AFTER A WRITE
6377                                          ;CHECK DATA COMMAND
6378                                          ;WORD NO CONTAINS THE WORD
6379                                          ;NUMBER THAT GOT CHANGED
6380  032064  000207                          RTS     PC              ;RETURN TO COMPARISON SUBROUTINE
6381
6382  032066  000240                  13$:    NOP                     ;ONLY A BRANCH POINT
6383
6384
6385
```

```
                                ;*************************************************************
                                ;*TEST 74       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
                                ;*************************************************************
032070  000004                TST74:  SCOPE

032072  012706  001000                MOV     #STACK,SP       ;RESET STACK

032076  012737  000074  002020        MOV     #TTNO,@#TSTNM   ;THIS SAVES TEST NUMBER

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

032104  012746  125252                MOV     #125252,-(SP)   ;DATA TO BE READ
032110  012705  000400                MOV     #256.,R5        ;COUNTER
032114  012700  047070                MOV     #DISK,R0        ;START OF SIMULATED DISK DATA
032120  011620                1$:     MOV     (SP),(R0)+      ;MOVE IN DATA ON TO SIMULATED DISK
032122  005305                        DEC     R5              ;COUNT
032124  001375                        BNE     1$              ;BRANCH IF 256 NOT COMPLETE
032126  005726                        TST     (SP)+           ;UNDO -(SP)
032130  012705  000021                MOV     #17.,R5         ;2 ECC WORDS
                                                              ;1 DATA GAP
                                                              ;14 TOLERANCE GAP
032134  005020                2$:     CLR     (R0)+           ;CLEAR ECC, DATA GAP, AND
032136  005305                        DEC     R5              ;TOLERANCE GAP
032140  001375                        BNE     2$              ;BRANCH IF NOT COMPLETE


                                ;THESE ARE TO SETUP FOR DISKLESS USE ONLY

032142  012737  000000  045152        MOV     #0!0,@#CYL ;16 BITS PER WORD
                                                              ;CYLINDER 0, FORMAT 16 BITS
032150  112737  000001  045155        MOVB    #1,@#SECOTR+1  ;TRACK 1
032156  112737  000000  045154        MOVB    #0,@#SECOTR     ;SECTOR 0
032164  012737  000000  045156        MOV     #0,@#KEY1       ;KEY1=0
032172  012737  000000  045160        MOV     #0,@#KEY2       ;KEY2=0
032200  012737  000004  045232        MOV     #4.,@#DAWORD    ;NO. OF DATA WORDS
032206  005037  045162                CLR     @#X             ;THIS IS A READ COMMAND
032212  004537  041730                JSR     R5,@#CRC        ;GO TO CALCULATE CRC
032216  045152                        CYL
032220  047052                        WCRC


                                ;THESE ARE REGULAR SETUPS FOR RH11 AND "REINTO"

032222  004737  040522                JSR     PC,@#CLDISK     ;SETUP GENERAL REGISTERS
032226  012777  177770  147370        MOV     #-4.-4.,@RHWC   ;4. DATA 4 HEADER WORDS
032234  012777  003134  147364        MOV     #REINTO,@RHBA   ;STARTING ADDRESS OF READ BUFFER
```

```
6440  032242  112746  000000          MOVB    #0,     -(SP)   ;IN LOWER BYTE GET SECTOR
6441  032246  112766  000001  000001  MOVB    #1,     1(SP)   ;GET TRACK IN HIGHER BYTE
6442  032254  012677  147356          MOV     (SP)+,  @RHDST  ;TRACK/SECTOR IN RHDST
6443  032260  012777  014000  147354  MOV     #FMT22!ECI,@RHOF ;16 BITS PER WORD
6444                                                          ;ECC CORRECTION INHIBIT
6445                                                          ;BECAUSE ECC IS NOT GOING
6446                                                          ;TO BE CHECKED
6447  032266  005077  147352          CLR     @RHCA           ;CYLINDER 0
6448
6449  032272  004737  040556          JSR     PC,@#CHECKT     ;CHECK THAT DVA,RDY,DPR,DRY = 1
6450  032276  104400  057176          TYPE    ,CPHALT         ;AND THAT NO OTHERS = 1.  CANNOT CON-
6451                                                          ;TINUE TESTING IF BOTH AREN'T TRUE
6452  032302  000000                  HALT                    ;STOP THE TEST
6453
6454  032304  013711  002052          MOV     @#REFOR,@R1     ;READ HEADER AND DATA=72
6455  032310  005037  001774          CLR     @#ERFLG$        ;CLEAR ERROR FLAG
6456  032314  004737  045026          JSR     PC,@#COMHD      ;READ HEADER AND DATA
6457
6458
6459                                  ;IF THE PROGRAM COMES BACK HERE WITHOUT ERROR PRINTOUTS
6460                                  ;FROM THE "COMHD" ROUTINE THAT MEANS SECTOR GAP,
6461                                  ;FIRST SYNC, HEADER, HEADER CRC, HEADER GAP AND
6462                                  ;SYNC BYTE HAVE GONE BY AND SYNCS WERE CORRECTLY
6463                                  ;DETECTED.
6464
6465
6466                                  ;RHMC IS CHECKED TO BE = 0 AFTER THE READ OPERATION
6467
6468  032320  017767  147300  146600  MOV     @RHMC,$BDDAT    ;LOAD AND TEST RHMC
6469  032326  001401                  BEQ     20$             ;SHOULD = 0
6470  032330  104040                  ERROR   40              ;RHMC DOES NOT = 0 AFTER A READ
6471                                                          ;HEADER AND DATA
6472
6473                                  ;HEADER AND DATA ARE TO BE CHECKED.
6474                                  ;IN CHECKING READ DATA THE WRITE FROM BUFFER
6475                                  ;"WRFROM" IS FILLED WITH EXPECTED DATA AND
6476                                  ;COMPARISONS ARE MADE.
6477
6478  032332  005737  001774  20$:   TST     @#ERFLG$        ;ANY ERRORS ALREADY THERE
6479  032336  001055                  BNE     TST75           ;BRANCH IF YES
6480  032340  004737  040746          JSR     PC,@#CHECKE     ;CHECK THAT BITS = 1
6481  032344  104400  057176          TYPE    ,CPHALT         ;CANNOT CONTINUE TESTING IF THEY DON'T
6482  032350  000000                  HALT                    ;STOP THE TEST
6483  032352  012700  002070          MOV     #WRFROM,R0      ;GETTING READY TO FILL EXPECTED DATA
6484  032356  012720  000000          MOV     #0,(R0)+        ;CYLINDER 0
6485  032362  112746  000000          MOVB    #0,-(SP)        ;IN LOWER BYTE GET SECTOR
6486  032366  112766  000001  000001  MOVB    #1,1(SP)        ;GET TRACK IN HIGHER BYTE
6487  032374  012620                  MOV     (SP)+,(R0)+     ;GET TRACK/SECTOR IN BUFFER
6488  032376  012720  000000          MOV     #0,(R0)+        ;KEY1 IN BUFFER
6489  032402  012720  000000          MOV     #0,(R0)+        ;KEY2 IN BUFFER
6490  032406  012701  000400          MOV     #256.,R1        ;DATA WORD COUNTER
6491  032412  012702  125252          MOV     #125252,R2              ;DATA
6492  032416  010220          3$:     MOV     R2,(R0)+        ;DATA INTO BUFFER
6493  032420  005301                  DEC     R1              ;COUNT
```

```
6494  032422  001375                          BNE     3$              ;BRANCH IF 256 NOT DONE
6495
6496                                          ;NOW READ DATA BUFFER WILL BE CHECKED
6497
6498  032424  004037  041416                  JSR     R0,@#COMPAR     ;CHECK
6499  032430  002070                          WRFROM                  ;GOOD BUFFER
6500  032432  003134                          REINTO                  ;TEST BUFFER
6501  032434  000010                          4+4.                    ;NUMBER OF WORDS CHECKED
6502  032436  032444                          4$                      ;RETURN POINT FOR ERROR HEADER
6503  032440  032450                          5$                      ;RETURN POINT FOR ERROR DATA
6504  032442  032454                          6$                      ;RETURN FOR GOOD COMPARISON
6505  032444  104004                  4$:     ERROR   4               ;READ NEXT ERROR
6506  032446  000207                          RTS     PC              ;RETURN TO "COMPAR"
6507  032450  104005                  5$:     ERROR   5               ;WORD NOS 1 TO 4 ARE
6508                                                                  ;HEADER WORDS
6509                                                                  ;5 TO 260 ARE DATA WORDS
6510  032452  000207                          RTS     PC              ;RETURN TO "COMPAR"
6511
6512
6513                                          ;NOW SEE THAT FORMAT ERROR BIT GOT SET
6514
6515  032454  004737  040126          6$:     JSR     PC,@#PUTREG     ;SAVE REGISTERS
6516
6517  032460  022737  100020  001704          CMP     #FER!DCK,@#ER1  ;FORMAT ERROR SHOULD BE SET
6518  032466  001401                          BEQ     TST75           ;BRANCH IF GOOD
6519  032470  104020                          ERROR   20              ;A 16 BIT PER WORD READ WAS ATTEMPTED
6520                                                                  ;WHEN THE DISK HAD
6521                                                                  ;THE FORMAT BIT=0= 18 BITS PER
6522                                                                  ;WORD THE READ WAS
6523                                                                  ;COMPLETED BUT ERROR REG
6524                                                                  ;WAS NOT RIGHT
6525                                                                  ;NOTE DCK WILL BE SET BECAUSE
6526                                                                  ;ECC HAS NOT BEEN GENERATED
6527
6528
```

```
                                ;*****************************************************************
                                ;*TEST 75       ERROR REGISTER #1-BIT 4 -FORMAT ERROR

                                ;*      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.

                                ;*****************************************************************
032472  000004                  TST75:  SCOPE
                                        ;NOW A WRITE DATA WILL BE ATTEMPTED WITH
                                        ;WRONG FORMAT BIT

032474  012706  001000                  MOV     #STACK,SP       ;RESET STACK

032500  012737  000075  002020          MOV     #TTNO,@#TSTNM   ;THIS SAVES TEST NUMBER

032506  012737  177777  045266          MOV     #-1,@#NOSYNC    ;SET FLAG SO THAT DATA SYNC
                                                                ;AND DATA IS NOT READ
032514  004037  040440          FRMAT1: JSR     R0,@#CLAREA     ;CLEAR SIMULATED DISK
032520  047070                          .WORD   DISK            ;FROM
032522  050114                          .WORD   TOLGAP+16       ;TO
032524  000000                          .WORD   0               ;DATA
                                ;THESE ARE SETUP FOR DISKLESS USE ONLY
032526  005037  045152                  CLR     @#CYL           ;CYLINDER 0, FORMAT 18 BIT WORDS
032532  105037  045155                  CLRB    @#SECOTR+1      ;TRACK 0
032536  105037  045154                  CLRB    @#SECOTR        ;SECTOR 0
032542  005037  045156                  CLR     @#KEY1          ;KEY1 0
032546  005037  045160                  CLR     @#KEY2          ;KEY2 0
032552  012737  000004  045220          MOV     #4,@#NOWORD     ;NO OF DATA WORDS
032560  012737  000001  045162          MOV     #1,@#X          ;WRITE DATA
032566  004537  041730                  JSR     R5,@#CRC        ;GO TO CALCULATE CRC
032572  050274                          WCYL
032574  050304                          GCRC

                                ;THESE AER REGULAR SETUPS

032576  004037  040440                  JSR     R0,@#CLAREA     ;FILL WRITE FROM BUFFER WITH 125252
032602  002070                          WRFROM                  ;FROM
032604  002076                          WRFROM+6                ;TO
032606  125252                          125252                  ;DATA
032610  004737  040522                  JSR     PC,@#CLDISK     ;SETUP GENERAL REGISTERS
032614  012777  177774  147002          MOV     #-4,@RHWC       ;256 DATA WORDS
032622  012777  002070  146776          MOV     #WRFROM,@RHBA   ;STARTING ADDRESS OF WRITE BUFFER
032630  005077  147002                  CLR     @RHDST          ;TRACK=0 SECTOR=0
032634  012777  010000  147000          MOV     #FMT22,@RHOF    ;16 BITS PER WORD FORMAT
032642  005077  146776                  CLR     @RHCA           ;CYLINDER 0
032646  004737  040556                  JSR     PC,@#CHECKT     ;CHECK THAT DVA,RDY,DPR,DRY = 1
032652  104400  057176                  TYPE    ,CPHALT         ;AND THAT NO OTHERS = 1.  CANNOT CON-
                                                                ;TINUE TESTING IF BOTH AREN'T TRUE
032656  000000                          HALT                    ;STOP THE TEST
032660  013711  002044                  MOV     @#WRIDAT,@R1    ;WRITE DATA=60
032664  005037  001774                  CLR     @#ERFLGS        ;CLEAR ERROR FLAG
```

```
6583  032670  004737  045026                    JSR     PC,@#COMHD      ;WRITE DATA
6584                                    ;IF THE PROGRAM COMES BACK HERE WITHOUT ERROR PRINTOUTS
6585                                    ;FROM THE "COMHD" ROUTINE IT MEANS SECTOR GAP, SYNC BYTE
6586                                    ;HEADER, HEADER CRC, HEADER GAP AND SYNC BYTE HAVE GONE BY
6587                                    ;AND SYNCS WERE CORRECTLY DETECTED
6588                                    ;DATA IS TO BE CHECKED
6589  032674  004737  040126                    JSR     PC,@#PUTREG     ;SAVE REGISTERS
6590  032700  005737  001774                    TST     @#ERFLGS        ;HAS ANY ERRORS OCCURED?
6591  032704  001041                            BNE     4$              ;BRANCH IF YES
6592  032706  012700  000000                    MOV     #0,R0           ;GOOD DATA
6593  032712  012701  047070                    MOV     #DISK,R1        ;DATA WRITTEN INTO "DISK"
6594  032716  012702  000004                    MOV     #4,R2           ;COUNTER
6595  032722  012737  000005  045272  1$:       MOV     #5,@#ERWORD     ;FOR ERROR WORD
6596  032730  020021                            CMP     R0,(R1)+        ;COMPARE GOOD DATA WITH DATA ON DISK
6597  032732  001424                            BEQ     3$              ;BRANCH IF GOOD
6598  032734  010037  001124                    MOV     R0,@#SGDDAT     ;GOOD DATA
6599  032740  014137  001126                    MOV     -(R1),@#SBDDAT  ;BAD DATA
6600  032744  160237  045272                    SUB     R2,@#ERWORD     ;ERROR WORD NO
6601  032750  005737  001774                    TST     @#ERFLGS        ;ANY ERRORS ALREADY THERE?
6602  032754  001002                            BNE     2$              ;BRANCH IF YES
6603  032756  104004                            ERROR   4               ;ERROR ON WRITE DATA COMMAND
6604                                                                    ;ON A WRITE DATA WITH
6605                                                                    ;WRONG FORMAT NO DATA
6606                                                                    ;SHOULD BE WRITTEN
6607                                                                    ;WORD NO GIVES WORD IN ERROR
6608  032760  000401                            BR      5$              ;BRANCH TO AVOID PRINTING NEXT ERROR
6609  032762  104005                    2$:     ERROR   5
6610  032764  005721                    5$:     TST     (R1)+           ;UNDO -(R1) FOR BAD DATA
6611  032766  013746  177570                    MOV     @#SWR,-(SP)     ;GET SWITCH SETTING
6612  032772  042716  177177                    BIC     #177177,(SP)    ;KEEP ONLY SWITCH 7 AND 8
6613  032776  022726  000200                    CMP     #SW07,(SP)+     ;IS 7 SET AND 8 RESET.
6614  033002  001402                            BEQ     4$              ;BRANCH IF YES
6615  033004  005302                    3$:     DEC     R2              ;IF NOT COUNT 256 WORDS
6616  033006  001345                            BNE     1$              ;BRANCH IF 256 NOT DONE
6617
6618                                    ;NOW CHECK TO SEE THAT FORMAT ERROR BIT GOT SET
6619
6620  033010  022737  000020  001704  4$:       CMP     #FER,@#ER1      ;FORMAT ERROR SHOULD BE SET
6621  033016  001401                            BEQ     TST76           ;BRANCH IF GOOD
6622  033020  104020                            ERROR   20              ;A 16 BIT PER WORD WRITE DATA
6623                                                                    ;WAS ATTEMPTED WHEN THE DISK
6624                                                                    ;HAD THE FORMAT BIT =0=18
6625                                                                    ;BITS PER WORD THE WRITE
6626                                                                    ;WAS CORRECTLY ABORTED
6627                                                                    ;BUT ERROR REG. 1 WAS WRONG
6628
```

```
                                ;*****************************************************************
                                ;*TEST 76       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
                                ;
                                ;*****************************************************************
033022  000004                  TST76:  SCOPE
                                ;
033024  012706  001000                  MOV     #STACK,SP       ;RESET STACK
033030  012737  000076  002020          MOV     #TTNO,@#TSTNM   ;THIS SAVES TEST NUMBER
033036  004737  040522                  JSR     PC,@#CLDISK     ;CLEAR DISK
033042  004737  040556                  JSR     PC,@#CHECKT     ;CHECK THAT DVA,RDY,DPR,DRY = 1
033046  104400  057176                  TYPE    ,CPHALT         ;AND THAT NO OTHERS = 1.  CANNOT CON-
                                                                ;TINUE TESTING IF BOTH AREN'T TRUE
033052  000000                          HALT                    ;STOP THE TEST
033054  012700  001644                  MOV     #RHCA,R0
033060  012005                  1$:     MOV     (R0)+,R5        ;R5 HAS ADDRESS OF REG. UNDER TEST
033062  052777  000040  146540  2$:     BIS     #CLR,@RHCS2
033070  013777  001762  146532          MOV     @#UNIT,@RHCS2   ;REINSTATE UNIT NO.
                                ;
                                ;SET UP FOR AN OPERATION (WRITE HEADER AND DATA)
                                ;
033076  013777  002046  146526          MOV     @#WRIFOR,@RHCS1 ;WRITE HEADER AND DATA=62
                                                                ;IN RHCS1
033104  012777  177766  146512          MOV     #-10.,@RHWC     ;10 WORDS
033112  012777  002070  146506          MOV     #WRFROM,@RHBA   ;BUS ADDRESS = WRFROM
033120  012777  000010  146510          MOV     #10,@RHDST      ;DESIRED TRACK=0, SECTOR=10
033126  052777  000010  146474          BIS     #BAI,@RHCS2     ;BUS ADDRESS INCREMENT INHIBIT
033134  012777  010000  146500          MOV     #FMT22,@RHOF    ;FORMAT 16 BIT WORDS
033142  005077  146476                  CLR     @RHCA           ;CYLINDER =0
                                ;
                                ;SAVE REGISTERS
                                ;
033146  004037  041214                  JSR     R0,@#SAVER      ;SAVE
033152  001632                          RHCS1                   ;FROM
033154  003134                          REINTO                  ;TO
033156  000016                          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
                                        ;
033160  012777  000001  146464          MOV     #DMD,@RHMR      ;SET DIAGNOSTIC MODE
033166  005277  146440                  INC     @RHCS1          ;GO TO RHCS1 WITH 62
033172  012715  177672                  MOV     #177672,@RS     ;TRY WRITING ALL BITS EXCEPT
                                                                ;GO, RMP, IE
033176  052737  000001  003154          BIS     #DMD,@#REINTO+20 ;SET DMD IN SAVED REGISTER RHMR
033204  052737  000004  003136          BIS     #RMR,@#REINTO+2 ;SET RMR IN SAVED REG. RHER1
033212  042737  000200  003156          BIC     #DRY,@#REINTO+22        ;CLEAR DRY IN RHDS1
033220  052737  040000  003156          BIS     #ERR,@#REINTO+22        ;SET ERR IN RHDS1
033226  052737  000001  003134          BIS     #GO,@#REINTO    ;SET GO IN SAVED REG. RHCS1
033234  042737  000200  003134          BIC     #RDY,@#REINTO   ;CLEAR RDY BIT
                                        ;
                                        ;AFTER AN ATTEMPT TO WRITE INTO A REGISTER
                                        ;SAVE REGISTERS AGAIN SO THAT COMPARES CAN BE DONE
                                        ;
033242  004037  041214                  JSR     R0,@#SAVER      ;SAVE
033246  001632                          RHCS1                   ;FROM
033250  002070                          WRFROM                  ;TO
033252  000016                          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
033254  113737  003153  002107          MOVB    @#REINTO+17,@#WRFROM+17;SAVE UPPER RHAS
                                        ;
                                        ;COMPARE REGISTERS BEFORE ATTEMPTED WRITE WITH AFTER
                                        ;
033262  004037  041416                  JSR     R0,@#COMPAR     ;COMPAR
033266  003134                          REINTO                  ;GO BUFFER
033270  002070                          WRFROM                  ;TEST BUFFER
033272  000016                          14.                     ;NUMBER
033274  033302                          4$                      ;RETURN FOR ERROR
033276  033302                          4$                      ;SAME
033300  033322                          5$                      ;RETURN FOR GOOD COMPARISON
033302  013705  045272          4$:     MOV     @#ERWORD,R5     ;GETTING READY TO INDEX
033306  060505                          ADD     R5,R5           ;DOUBLE ERROR WORD
033310  016537  001630  040172          MOV     RHCS1-2(R5),@#REGADR ;FAILING REG. ADDRESS
033316  104001                          ERROR   1               ;CONTENTS OF REGISTER
033320  000207                          RTS     PC              ;CHANGED WITH
                                                                ;AN ATTEMPT TO WRITE
                                                                ;DURING AN OPERATION
                                        ;THE FOLLOWING CLEAR MAY SET THE ATA BIT BECAUSE GO IS HIGH
                                        ;
033322  004737  040522          5$:     JSR     PC,@#CLDISK     ;CLEAR DISK
```

```
                                ;*****************************************************************
                                ;*TEST 77       MAKE CURRENT CYLINDER = 1
                                ;*****************************************************************
033326  000004                  TST77:  SCOPE
033330  012706  001000                  MOV     #STACK,SP       ;RESET STACK
033334  004737  040522                  JSR PC,@#CLDISK         ;INIT DRIVE
033340  012777  000001  146304          MOV     #DMD,@RHMR      ;SET DIAGNOSTIC MODE
033346  004037  043074                  JSR     R0,@#MAKECYL    ;SUBROUTINE TO GIVE A SEEK
                                                                ;COMMAND FOLOWED BY A INIT
                                                                ;THIS SHUOLD CHANGE RHCC
033352  000001                          1                       ;CHANGE RHCC TO 1


                                ;*****************************************************************
                                ;*TEST 100      ERROR REG1 - BIT #7 - HEADER COMPARE 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=1, TRACK=1, SECTOR=1, KEY1=1, KEY2=1
                                ;*      REINTO BUFFER IS FILLED WITH 0
                                ;*      WRFROM IS FILLED WITH 10000,401,1,1, AND ALL 177400
                                ;*      AFTER THE READ THE REINTO BUFFER IS EXPECTED TO
                                ;*      HAVE WHAT IS IN WRFROM - 10000,401,1,1 AND ALL 177400

                                ;*****************************************************************
033354  000004                  TST100: SCOPE


033356  012706  001000                  MOV     #STACK,SP       ;RESET STACK
033362  012737  000100  002020          MOV     #TTNO,@#TSTNM   ;THIS SAVES TEST NUMBER

033370  005037  045266                  CLR     @#NOSYNC        ;SET FLAG SO THAT DATA SYNC
                                                                ;AND DATA IS  READ
                                ;FILL SIMULATED DISK

033374  004737  042204                  JSR     PC,@#SETDSK     ;SET UP SIMULATED DISK

                                ;FILL REINTO BUFFER WITH 0

033400  004037  040440                  JSR     R0,@#CLAREA     ;FILL REINTO BUFFER
033404  003134                          REINTO                  ;FROM LOCATION
033406  004134                          REINTO+<256.*2>         ;TO LOCATION
033410  000000                          0                       ;DATA

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

033412  012700  002070                  MOV     #WRFROM,R0
033416  012720  010000                  MOV     #FMT22,(R0)+    ;10000 INTO WRFROM
033422  012720  000401                  MOV     #401,(R0)+      ;401=TRACK1,SECTOR1
033426  012720  000001                  MOV     #1,(R0)+        ;1 INTO WRFROM+
```

```
6785  033432  012720  000001                  MOV     #1,(R0)+        ;1 INTO WRFROM+6
6786
6787                                          ;FILL ALL 0
6788
6789  033436  004037  040440                  JSR     R0,@#CLAREA     ;FILL WRFROM
6790  033442  002100                          WRFROM+10               ;FROM
6791  033444  003070                          WRFROM+<256.*2>         ;TO
6792  033446  177400                          177400                  ;DATA
6793
6794                                          ;NOW GIVE A READ HEADER AND DATA COMMAND
6795                                          ;CYLINDER=1
6796                                          ;TRACK = 1
6797                                          ;SECTOR = 1
6798
6799  033450  004037  042332                  JSR     R0,@#HCCRCE
6800  033454  000072                          72                      ;READ HEADER AND DATA
6801  033456  000001                          1                       ;CYLINDER
6802  033460  000001                          1                       ;SECTOR
6803  033462  000001                          1                       ;TRACK
6804  033464  177400                          -256.                   ;WORD COUNT
6805  033466  003134                          REINTO                  ;RHBA BUFFER
6806  033470  000000                          0                       ;READ
6807
6808  033472  000001                          1                       ;HEADER COMPARE
6809  033474  000240                  1$:     NOP                     ;RETURN POINT FROM HCCRCE
6810
6811
```

```
6812
6813                                    ;*****************************************************************
6814                                    ;*TEST 101      MAKE CURRENT CYLINDER = 0
6815                                    ;*****************************************************************
6816  033476  000004                    TST101: SCOPE
6817  033500  012706  001000                    MOV     #STACK,SP       ;RESET STACK
6818  033504  004737  040522                    JSR PC,@#CLDISK         ;INIT DRIVE
6819  033510  012777  000001  146134            MOV     #DMD,@RHMR      ;SET DIAGNOSTIC MODE
6820  033516  004037  043074                    JSR     R0,@#MAKECYL    ;SUBROUTINE TO GIVE A SEEK
6821                                                                    ;COMMAND FOLOWED BY A INIT
6822                                                                    ;THIS SHUOLD CHANGE RHCC
6823  033522  000000                            0                       ;CHANGE RHCC TO 0
6824
6825
6826
6827                                    ;*****************************************************************
6828                                    ;*TEST 102      ERROR REG1 - BIT #7 - HEADER COMPARE ERROR
6829
6830                                    ;*      THE SIMULATED DISK IS SET TO READ CYLINDER=0, TRACK=1
6831                                    ;*      SECTOR=1, KEYS=1, 256 WORDS OF 177400
6832                                    ;*      A READ HEADER AND DATA COMMAND IS GIVEN TO READ
6833                                    ;*      CYLINDER=0, TRACK=0, SECTOR=1, KEY1=1, KEY2=1
6834                                    ;*      REINTO BUFFER IS FILLED WITH 0
6835                                    ;*      WRFROM IS FILLED WITH 10000,401,1,1,1, AND ALL 177400
6836                                    ;*      AFTER THE READ THE REINTO BUFFER IS EXPECTED TO
6837                                    ;*      HAVE WHAT IS IN WRFROM - 10000,401,1,1 AND ALL 177400
6838
6839                                    ;*****************************************************************
6840  033524  000004                    TST102: SCOPE
6841
6842
6843  033526  012706  001000                    MOV     #STACK,SP       ;RESET STACK
6844  033532  012737  000102  002020            MOV     #TTNO,@#TSTNM   ;THIS SAVES TEST NUMBER
6845
6846  033540  005037  045266                    CLR     @#NOSYNC        ;SET FLAG SO THAT DATA SYNC
6847                                                                    ;AND DATA IS  READ
6848                                            ;FILL SIMULATED DISK
6849
6850  033544  004737  042204                    JSR     PC,@#SETDSK     ;SET UP SIMULATED DISK
6851
6852                                            ;FILL REINTO BUFFER WITH 0
6853
6854  033550  004037  040440                    JSR     R0,@#CLAREA     ;FILL REINTO BUFFER
6855  033554  003134                            REINTO                  ;FROM LOCATION
6856  033556  004134                            REINTO+<256.*2>         ;TO LOCATION
6857  033560  000000                            0                       ;DATA
6858
6859                                            ;FILL WRFROM WITH 10000,401,1,1, AND ALL 177400
6860
6861  033562  012700  002070                    MOV     #WRFROM,R0
6862  033566  012720  010000                    MOV     #FMT22,(R0)+    ;10000 INTO WRFROM
6863  033572  012720  000401                    MOV     #401,(R0)+      ;401=TRACK1,SECTOR1
6864  033576  012720  000001                    MOV     #1,(R0)+        ;1 INTO WRFROM+
6865  033602  012720  000001                    MOV     #1,(R0)+        ;1 INTO WRFROM+6
```

```
6866
6867                                            ;FILL ALL 0
6868
6869  033606  004037  040440                    JSR     R0,@#CLAREA     ;FILL WRFROM
6870  033612  002100                            WRFROM+10               ;FROM
6871  033614  003070                            WRFROM+<256.*2>         ;TO
6872  033616  177400                            177400                  ;DATA
6873
6874                                            ;NOW GIVE A READ HEADER AND DATA COMMAND
6875                                            ;CYLINDER=0
6876                                            ;TRACK = 0
6877                                            ;SECTOR = 1
6878
6879  033620  004037  042332                    JSR     R0,@#HCCRCE
6880  033624  000072                            72                      ;READ HEADER AND DATA
6881  033626  000000                            0                       ;CYLINDER
6882  033630  000001                            1                       ;SECTOR
6883  033632  000000                            0                       ;TRACK
6884  033634  177400                            -256.                   ;WORD COUNT
6885  033636  003134                            REINTO                  ;RHBA BUFFER
6886  033640  000000                            0                       ;READ
6887
6888  033642  000001                            1                       ;HEADER COMPARE
6889  033644  000240                    1$:     NOP                     ;RETURN POINT FROM HCCRCE
6890
6891
```

```
6892
6893                                    ;*****************************************************************
6894                                    ;*TEST 103      MAKE CURRENT CYLINDER = 1
6895                                    ;*****************************************************************
6896  033646  000004                    TST103: SCOPE
6897  033650  012706  001000                    MOV     #STACK,SP       ;RESET STACK
6898  033654  004737  040522                    JSR PC,@#CLDISK         ;INIT DRIVE
6899  033660  012777  000001  145764            MOV     #DMD,@RHMR      ;SET DIAGNOSTIC MODE
6900  033666  004037  043074                    JSR     R0,@#MAKECYL    ;SUBROUTINE TO GIVE A SEEK
6901                                                                    ;COMMAND FOLOWED BY A INIT
6902                                                                    ;THIS SHUOLD CHANGE RHCC
6903  033672  000001                    1                            ;CHANGE RHCC TO 1
6904
6905
6906
6907                                    ;*****************************************************************
6908                                    ;*TEST 104      ERROR REG.1 - BIT #7 - HEADER COMPARE ERROR
6909
6910                                    ;*      THE SIMULATED DISK IS SET UP FOR CYLINDER=0, TRACK=1
6911                                    ;*      SECTOR=1, KEYS=1, 256 WORDS OF 177400
 312                                    ;*      A WRITE DATA COMMAND IS GIVEN TO WRITE CYLINDER=1
6913                                    ;*      TRACK=1, SECTOR=1, KEY1=1, KEY2=1
6914                                    ;*      WRFROM BUFFER IS FILLED WITH 125252
6915                                    ;*      REINTO BUFFER IS FILLED WITH 177400
6916                                    ;*      AFTER THE WRITE COMMAND THE DISK IS EXPECTED TO
6917                                    ;*      HAVE 177400
6918
6919                                    ;*****************************************************************
6920  033674  000004                    TST104: SCOPE
6921
6922
6923  033676  012706  001000                    MOV     #STACK,SP       ;RESET STACK
6924  033702  012737  000104  002020            MOV     #TTNO,@#TSTNM   ;THIS SAVES TEST NUMBER
6925
6926  033710  012737  177777  045266            MOV     #-1,@#NOSYNC    ;SET FLAG SO THAT DATA SYNC
6927                                                                    ;AND DATA IS NOT READ
6928                                    ;FILL SIMULATED DISK
6929
6930  033716  004737  042204                    JSR     PC,@#SETDSK     ;SETUP SIMULATED DISK
6931
6932                                    ;FILL WRFROM WITH 125252
6933
6934  033722  004037  040440                    JSR     R0,@#CLAREA     ;FILL WRFROM BUFFER
6935  033726  002070                    WRFROM                          ;FROM LOCATION
6936  033730  003070                    WRFROM+<256.*2>                 ;TO LOCATION
6937  033732  125252                    125252                          ;DATA
6938
6939                                    ;FILL REINTO WITH 256 WORDS OF 177400
6940                                    ;THIS IS WHAT IS EXPECTED TO BE ON DISK EVEN AFTER
6941                                    ;AN ATTEMPT TO WRITE 125252
6942
6943  033734  004037  040440                    JSR     R0,@#CLAREA     ;FILL REINTO BUFFER
6944  033740  003134                    REINTO                          ;FROM LOCATION
6945  033742  004134                    REINTO+<256.*2>                 ;TO
```

```
6946  033744  177400                          177400
6947
6948                                          ;NOW GIVE A WRITE DATA COMMAND
6949                                          ;CYLINDER = 1
6950                                          ;TRACK = 1
6951                                          ;SECTOR = 1
6952
6953  033746  004037  042332                  JSR     R0,@#HCCRCE
6954  033752  000060                          60                      ;WRITE DATA
6955  033754  000001                          1                       ;CYLINDER
6956  033756  000001                          1                       ;SECTOR
6957  033760  000001                          1                       ;TRACK
6958  033762  177400                          -256.                   ;WORD COUNT
6959  033764  002070                          WRFROM                  ;RHBA BUFFER
6960  033766  000001                          1                       ;WRITE
6961
6962  033770  000001                          1                       ;HEADER COMPARE
6963  033772  000240                  15:     NOP                     ;RETURN POINT FROM HCCRCE
6964
```

```
                                        ;*****************************************************************
                                        ;*TEST 105      MAKE CURRENT CYLINDER = 0
                                        ;*****************************************************************
034774  000004                          ↑ST105: SCOPE
034776  012706  001000                          MOV     #STACK,SP       ;RESET STACK
035002  004737  040522                          JSR PC,@#CLDISK         ;INIT DRIVE
035006  012777  000001  145636                  MOV     #DMD,@RHMR      ;SET DIAGNOSTIC MODE
035014  004037  043074                          JSR     R0,@#MAKECYL    ;SUBROUTINE TO GIVE A SEEK
                                                                        ;COMMAND FOLOWED BY A INIT
                                                                        ;THIS SHUOLD CHANGE RHCC
035020  000000                                  0                       ;CHANGE RHCC TO 0


                                        ;*****************************************************************
                                        ;*TEST 106      ERROR REG.1 - BIT #7 - HEADER COMPARE 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=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

                                        ;*****************************************************************
034022  000004                          ↑ST106: SCOPE

034024  012706  001000                          MOV     #STACK,SP       ;RESET STACK
034030  012737  000106  002020                  MOV     #TTNO,@#TSTNM   ;THIS SAVES TEST NUMBER

034036  012737  177777  045266                  MOV     #-1,@#NOSYNC    ;SET FLAG SO THAT DATA SYNC
                                                                        ;AND DATA IS NOT READ
                                                ;FILL SIMULATED DISK

034044  004737  042204                          JSR     PC,@#SETDSK     ;SETUP SIMULATED DISK

                                                ;FILL WRFROM WITH 125252

034050  004037  040440                          JSR     R0,@#CLAREA     ;FILL WRFROM BUFFER
034054  002070                                  WRFROM                  ;FROM LOCATION
034056  003070                                  WRFROM+<256.*2>         ;TO LOCATION
034060  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

034062  004037  040440                          JSR     R0,@#CLAREA     ;FILL REINTO BUFFER
```

```
 7019  034066  003134                        REINTO                  ;FROM LOCATION
 7020  034070  004134                        REINTO+<256.*2>         ;TO
 7021  034072  177400                        177400
 7022
 7023                                        ;NOW GIVE A WRITE DATA COMMAND
 7024                                        ;CYLINDER = 0,
 7025                                        ;TRACK = 0
 7026                                        ;SECTOR = 1
 7027
 7028  034074  004037  042332                JSR     R0,@#HCCRCE
 7029  034100  000060                        60                      ;WRITE DATA
 7030  034102  000000                        0                       ;CYLINDER
 7031  034104  000001                        1                       ;SECTOR
 7032  034106  000000                        0                       ;TRACK
 7033  034110  177400                        -256.                   ;WORD COUNT
 7034  034112  002070                        WRFROM                  ;RHBA BUFFER
 7035  034114  000001                        1                       ;WRITE
 7036
 7037  034116  000001                        1                       ;HEADER COMPARE
 7038  034120  000240                1$:     NOP                     ;RETURN POINT FROM HCCRCE
 7039
```

```
                                ;*****************************************************************
                                ;*TEST 107      RHER1 - BIT #8 - CRC ERROR (READING)

                                ;*      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

                                ;*****************************************************************
034122  000004                 TST107: SCOPE

034124  012706  001000                 MOV     #STACK,SP       ;RESET STACK
034130  012737  000107  002020         MOV     #TTNO,@#TSTNM   ;THIS SAVES TEST NUMBER

034136  005037  045266                 CLR     @#NOSYNC        ;SET FLAG SO THAT DATA SYNC
                                                               ;AND DATA IS  READ
                                       ;FILL SIMULATED DISK

034142  004737  042204                 JSR     PC,@#SETDSK     ;SET UP SIMULATED DISK
034146  005137  047052                 COM     @#WCRC          ;CHANGE CRC TO GIVE WCRC

                                       ;FILL REINTO BUFFER WITH 0

034152  004037  040440                 JSR     R0,@#CLAREA     ;FILL REINTO BUFFER
034156  003134                         REINTO                  ;FROM LOCATION
034160  004134                         REINTO+<256.*2>         ;TO LOCATION
034162  000000                         0                       ;DATA

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

034164  012700  002070                 MOV     #WRFROM,R0
034170  012720  010000                 MOV     #FMT22,(R0)+    ;10000 INTO WRFROM
034174  012720  000401                 MOV     #401,(R0)+      ;401=TRACK1,SECTOR1
034200  012720  000001                 MOV     #1,(R0)+        ;1 INTO WRFROM+
034204  012720  000001                 MOV     #1,(R0)+        ;1 INTO WRFROM+6

                                       ;FILL ALL 0

034210  004037  040440                 JSR     R0,@#CLAREA     ;FILL WRFROM
034214  002100                         WRFROM+10               ;FROM
034216  003070                         WRFROM+<256.*2>         ;TO
034220  177400                         177400                  ;DATA

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

E01


```
7094
7095  034222  004037  042332          JSR     R0,@#HCCRCE
7096  034226  000072                  72                      ;READ HEADER AND DATA
7097  034230  000000                  0                       ;CYLINDER
7098  034232  000001                  1                       ;SECTOR
7099  034234  000001                  1                       ;TRACK
7100  034236  177400                  -256.                   ;WORD COUNT
7101  034240  003134                  REINTO                  ;RHBA BUFFER
7102  034242  000000                  0                       ;READ
7103
7104  034244  000000                  0                       ;CRC ERROR
7105  034246  000240          1$:     NOP                     ;RETURN POINT FROM HCCRCE
7106
7107
```

```
7108
7109
7110                                    ;*****************************************************************
7111                                    ;*TEST 110      RHER1 - BIT 8 - CRC ERROR (WRITING)
7112
7113                                    ;*      THE SIMULATED DISK IS SET UP FOR CYLINDER=0, TRACK=1
7114                                    ;*      SECTOR=1, KEYS=1, 256 WORDS OF 177400
7115                                    ;*      A WRITE DATA COMMAND IS GIVEN TO WRITE CYLINDER=0
7116                                    ;*      TRACK=1, SECTOR=1, KEY1=1, KEY2=1
7117                                    ;*      WRFROM BUFFER IS FILLED WITH 125252
7118                                    ;*      REINTO BUFFER IS FILLED WITH 177400
7119                                    ;*      AFTER THE WRITE COMMAND THE DISK IS EXPECTED TO
7120                                    ;*      HAVE 177400
7121
7122                                    ;*****************************************************************
7123  034250  000004                    TST110: SCOPE
7124
7125
7126  034252  012706  001000                    MOV     #STACK,SP       ;RESET STACK
7127  034256  012737  000110  002020            MOV     #TTNO,@#TSTNM   ;THIS SAVES TEST NUMBER
7128
7129  034264  012737  177777  045266            MOV     #-1,@#NOSYNC    ;SET FLAG SO THAT DATA SYNC
7130                                                                    ;AND DATA IS NOT READ
7131                                    ;FILL SIMULATED DISK
7132
7133  034272  004737  042204                    JSR     PC,@#SETDSK     ;SETUP SIMULATED DISK
7134  034276  005137  047052                    COM     @#WCRC          ;CHANGE CRC TO GIVE HCRC
7135
7136                                    ;FILL WRFROM WITH 125252
7137
7138  034302  004037  040440                    JSR     R0,@#CLAREA     ;FILL WRFROM BUFFER
7139  034306  002070                            WRFROM                  ;FROM LOCATION
7140  034310  003070                            WRFROM+<256.*2>         ;TO LOCATION
7141  034312  125252                            125252                  ;DATA
7142
7143                                    ;FILL REINTO WITH 256 WORDS OF 177400
7144                                    ;THIS IS WHAT IS EXPECTED TO BE ON DISK EVEN AFTER
7145                                    ;AN ATTEMPT TO WRITE 125252
7146
7147  034314  004037  040440                    JSR     R0,@#CLAREA     ;FILL REINTO BUFFER
7148  034320  003134                            REINTO                  ;FROM LOCATION
7149  034322  004134                            REINTO+<256.*2>         ;TO
7150  034324  177400                            177400
7151
7152                                    ;NOW GIVE A WRITE DATA COMMAND
7153                                    ;CYLINDER = 0,
7154                                    ;TRACK = 1
7155                                    ;SECTOR = 1
7156
7157  034326  004037  042332                    JSR     R0,@#HCCRCE
7158  034332  000060                            60                      ;WRITE DATA
7159  034334  000000                            0                       ;CYLINDER
7160  034336  000001                            1                       ;SECTOR
7161  034340  000001                            1                       ;TRACK
```

```
7162  034342  177400                      -256.                ;WORD COUNT
7163  034344  002070                      WRFROM               ;RHBA BUFFER
7164  034346  000001                      1                    ;WRITE
7165
7166  034350  000000                      0                    ;CRC ERROR
7167  034352  000240              1$:     NOP                  ;RETURN POINT FROM HCCRCE
7168
7169
```

```
                                ;*****************************************************************
                                ;*TEST 111      MAKE CURRENT CYLINDER = 410.
                                ;*****************************************************************
034354  000004                  TST111: SCOPE
034356  012706  001000                  MOV     #STACK,SP       ;RESET STACK
034362  004737  040522                  JSR PC,@#CLDISK         ;INIT DRIVE
034366  012777  000001  145256          MOV     #DMD,@RHMR      ;SET DIAGNOSTIC MODE
034374  004037  043074                  JSR     R0,@#MAKECYL    ;SUBROUTINE TO GIVE A SEEK
                                                                ;COMMAND FOLOWED BY A INIT
                                                                ;THIS SHUOLD CHANGE RHCC
034400  000632                          410.                    ;CHANGE RHCC TO 410.


                                ;*****************************************************************
                                ;*TEST 112      RHDS1 (BIT #10) - LAST SECTOR TRANSFERRED, 'LST'
                                ;*      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.
                                ;*****************************************************************
034402  000004                  TST112: SCOPE

034404  012706  001000                  MOV     #STACK,SP       ;RESET STACK
034410  012737  000112  002020          MOV     #TTNO,@#TSTNM   ;THIS SAVES TEST NUMBER

034416  004037  040440                  JSR     R0,@#CLAREA     ;CLEAR SIMULATED DISK
034422  047070                          .WORD   DISK            ;FROM
034424  050114                          .WORD   TOLGAP+16       ;TO
034426  000000                          .WORD   0               ;DATA

                                        ;THESE ARE SETUP FOR DISKLESS USE ONLY

034430  012737  010632  045152          MOV     #410.!FMT22,@#CYL;CYLINDER 410.
                                                                ;16 BITS PER WORD
034436  112737  000022  045155          MOVB    #18.,@#SECOTR+1;TRACK 18.
034444  112737  000025  045154          MOVB    #21.,@#SECOTR   ;SECTOR 21.
034452  005037  045156                  CLR     @#KEY1          ;KEY1 0
034456  005037  045160                  CLR     @#KEY2          ;KEY2 0
034462  012737  000400  045220          MOV     #256.,@#NOWORD  ;NO OF DATA WORDS
034470  012737  000001  045162          MOV     #1,@#X          ;WRITE DATA
034476  004537  041730                  JSR     R5,@#CRC        ;GO TO CALCULATE CRC
034502  045152                          CYL
034504  047052                          WCRC

                                ;THESE ARE REGULAR SETUPS

034506  004037  040440                  JSR     R0,@#CLAREA     ;FILL WRITE BUFFER WITH 377
034512  002070                          WRFROM                  ;FROM LOCATION
```

```
7224  034514  003070                        WRFROM+<256.*2>         ;TO LOCATION
7225  034516  000377                        377                     ;DATA
7226  034520  004737  040522                JSR     PC,@#CLDISK     ;SETUP GENERAL REGISTERS
7227  034524  012777  177400  145072        MOV     #-256.,@RHWC    ;256. DATA WORDS
7228  034532  012777  002070  145066        MOV     #WRFROM,@RHBA   ;STARTING ADDRESS OF WRITE BUFFER
7229  034540  012746  000025                MOV     #21.,-(SP)      ;SECTOR 21.
7230  034544  112766  000022  000001        MOVB    #18.,1(SP)      ;TRACK 18.
7231  034552  012677  145060                MOV     (SP)+,@RHDST    ;SECTOR 21. TRACK 18.
7232  034556  012777  010000  145056        MOV     #FMT22,@RHOF    ;16 BITS PER WORD FORMAT
7233  034564  012777  000632  145052        MOV     #410.,@RHCA     ;CYLINDER 410.
7234  034572  004737  040556                JSR     PC,@#CHECKT     ;CHECK THAT DVA,RDY,DPR,DRY = 1
7235  034576  104400  057176                TYPE    ,CPHALT         ;AND THAT NO OTHERS = 1.  CANNOT CON-
7236                                                                ;TINUE TESTING IF BOTH AREN'T TRUE
7237  034602  000000                        HALT                    ;STOP THE TEST
7238  034604  013711  002044                MOV     @#WRIDAT,@R1    ;WRITE DATA=60
7239  034610  005037  001774                CLR     @#ERFLG$        ;CLEAR ERROR FLAG
7240  034614  004737  045026                JSR     PC,@#COMMD      ;WRITE DATA
7241
7242
7243                                ;IF THE PROGRAM COMES BACK HERE WITHOUT ERROR PRINTOUTS
7244                                ;FROM THE "COMMD" ROUTINE IT MEANS SECTOR GAP, SYNC BYTE
7245                                ;HEADER, HEADER CRC, HEADER GAP AND SYNC BYTE HAVE GONE BY
7246                                ;AND SYNCS WERE CORRECTLY DETECTED, DATA IS TO BE CHECKED.
7247
7248  034620  004737  040126                JSR     PC,@#PUTREG     ;SAVE REGISTERS
7249  034624  005737  001774                TST     @#ERFLG$        ;HAVE ANY ERRORS OCCURED?
7250  034630  001062                        BNE     5$              ;BRANCH IF YES
7251  034632  012700  000377                MOV     #377,R0         ;GOOD DATA
7252  034636  012701  047070                MOV     #DISK,R1        ;DATA WRITTEN INTO "DISK"
7253  034642  012702  000400                MOV     #256.,R2        ;COUNTER
7254
7255  034646  012737  000401  045272  1$:   MOV     #256.+1,@#ERWORD;FOR ERROR WORD
7256  034654  020021                        CMP     R0,(R1)+        ;COMPARE GOOD DATA WITH DATA ON DISK
7257  034656  001424                        BEQ     3$              ;BRANCH IF GOOD
7258  034660  010037  001124                MOV     R0,@#$GDDAT     ;GOOD DATA
7259  034664  014137  001126                MOV     -(R1),@#$BDDAT  ;BAD DATA
7260  034670  160237  045272                SUB     R2,@#ERWORD     ;ERROR WORD NO
7261  034674  005737  001774                TST     @#ERFLG$        ;ANY ERRORS ALREADY THERE?
7262  034700  001002                        BNE     2$              ;BRANCH IF YES
7263  034702  104004                        ERROR   4               ;ERROR ON WRITE DATA COMMAND
7264  034704  000401                        BR      64$             ;BRANCH TO AVOID PRINTING NEXT ERROR
7265
7266  034706  104005                2$:     ERROR   5               ;WORD NO GIVES WORD IN ERROR
7267  034710  005721                64$:    TST     (R1)+           ;UNDO -(R1) FOR BAD DATA
7268  034712  013746  177570                MOV     @#SWR,-(SP)     ;GET SWITCH SETTING
7269  034716  042716  177177                BIC     #177177,(SP)    ;KEEP ONLY SWITCH 7 AND 8
7270  034722  022726  000200                CMP     #SW07,(SP)+     ;IS 7 SET AND 8 RESET
7271  034726  001402                        BEQ     4$              ;BRANCH OUT IF YES
7272  034730  005302                3$:     DEC     R2              ;IF NOT COUNT 256 WORDS
7273  034732  001345                        BNE     1$              ;BRANCH IF 256. NOT DONE
7274
7275
7276
7277  034734  013746  001724        4$:     MOV     @#DS1,-(SP)     ;GET RHDS1
```

```
7278  034740  042716  001000                BIC     #PROG,(SP)          ;CLEAR PROG
7279  034744  022726  002700                CMP     #LST!DPR!DRY!VV,(SP)+;IS LST HIGH ?
7280  034750  001412                        BEQ     5$                  ;BRANCH IF GOOD
7281  034752  013737  001654  040172        MOV     @#RHDS1,@#REGADR    ;FAILING REG. ADDRESS
7282  034760  012737  002700  001124        MOV     #LST!DPR!DRY!VV,@#$GDDAT ;GOOD DATA
7283  034766  013737  001724  001126        MOV     @#DS1,@#$BDDAT      ;BAD DATA
7284  034774  104001                        ERROR   1                   ;LST DID NOT SET AFTER
7285                                                                    ;LAST SECTOR ON LAST TRACK
7286                                                                    ;ON LAST CYLINDER WAS
7287                                                                    ;WRITTEN
7288                                                                    ;VV BIT #6 MAY OR MAY NOT BE HIGH
7289  034776  013737  001632  035006  5$:     MOV     @#RHCS1,@#6$
7290  035004  104424                        WAT
7291  035006  000000                6$:     0                   ;RHCS1 ADDRESS
7292  035010  000200                        RDY                         ;WAIT FOR READY
7293
7294
7295
```

```
                                        ;*****************************************************************
                                        ;*TEST 113      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
                                        ;
                                        ;*****************************************************************
035012  000004                  ST113:  SCOPE
035014  012706  001000                  MOV     #STACK,SP       ;RESET STACK
035020  012737  000113  002020          MOV     #TTNO,@#TSTNM   ;THIS SAVES TEST NUMBER
035026  004737  040522                  JSR     PC,@#CLDISK     ;INIT AND SET GENERAL REGISTERS
035032  004037  040440                  JSR     R0,@#CLAREA     ;CLEAR SIMULATED DISK
035036  047070                          .WORD   DISK            ;FROM
035040  050114                          .WORD   TOLGAP+16       ;TO
035042  000000                          .WORD   0                       ;DATA
                                        ;THESE ARE SETUP FOR DISKLESS USE ONLY
035044  012737  010632  045152          MOV     #410.!FMT22,@#CYL ;CYLINDER 410.
                                                                ;16 BITS PER WORD
035052  112737  000022  045155          MOVB    #18.,@#SECOTR+1 ;TRACK 18.
035060  112737  000025  045154          MOVB    #21.,@#SECOTR   ;SECTOR 21.
035066  005037  045156                  CLR     @#KEY1          ;KEY1 0
035072  005037  045160                  CLR     @#KEY2          ;KEY2 0
035076  012737  000400  045220          MOV     #256.,@#NOWORD  ;NO OF DATA WORDS
035104  012737  000001  045162          MOV     #1,@#X          ;WRITE DATA
035112  004537  041730                  JSR     R5,@#CRC        ;GO TO CALCULATE CRC
035115  045152                          CYL
035120  047052                          WCRC
                                        ;THESE ARE REGULAR SETUPS
035122  004037  040440                  JSR     R0,@#CLAREA     ;FILL WRITE BUFFER WITH 377
035126  002070                          WRFROM                  ;FROM
035130  003070                          WRFROM+<256.*2>         ;TO
035132  000377                          377                     ;DATA
035134  004737  040522                  JSR     PC,@#CLDISK     ;SETUP GENERAL REGISTERS
035140  012777  177272  144456          MOV     #-326.,@RHWC    ;326. DATA WORDS
035146  012777  002070  144452          MOV     #WRFROM,@RHBA   ;STARTING ADDRESS OF WRITE BUFFER
035154  012746  000025                  MOV     #21.,-(SP)      ;SECTOR 21.
035160  112766  000022  000001          MOVB    #18.,1(SP)      ;TRACK 18.
035166  012677  144444                  MOV     (SP)+,@RHDST    ;SECTOR 21. TRACK 18.
035172  012777  010000  144442          MOV     #FMT22,@RHOF    ;16 BITS PER WORD FORMAT
035200  012777  000632  144436          MOV     #410.,@RHCA     ;CYLINDER 410.
035206  004737  040556                  JSR     PC,@#CHECKT     ;CHECK THAT DVA,RDY,DPR,DRY = 1
035212  104400  057176                  TYPE    ,CPHALT         ;AND THAT NO OTHERS = 1.  CANNOT CON-
                                                                ;TINUE TESTING IF BOTH AREN'T TRUE
035216  000000                          HALT                    ;STOP THE TEST
```

```
7350  035220  013711  002044                  MOV     @#WRIDAT,@R1    ;WRITE DATA=60
7351  035224  005037  001774                  CLR     @#ERFLGS        ;CLEAR ERROR FLAG
7352
7353                                          ;THE REGISTERS WILL BE SAVED IN REINTO BUFFER
7354  035230  004037  041214                  JSR     R0,@#SAVER      ;SAVE
7355  035234  001624                          RH%C                    ;FROM
7356  035236  003134                          REINTO                  ;TO
7357  035240  000023                          19.                     ;NUMBER SAVED
7358
7359                                          ;GIVE WRITE DATA COMMAND
7360  035242  004737  045026                  JSR     PC,@#COMMD      ;WRITE DATA COMMAND
7361
7362                                          ;CHANGE SAVED REGISTERS TO EXPECTED VALUE
7363  035246  012737  177774  003134          MOV     #-4,@#REINTO    ;SAVED RH%C SHOULD BE=4
7364  035254  012737  003274  003136          MOV     #WRFROM+<2*256.>+<2*66.>,@#REINTO+2
7365                                                                  ;SAVED RHBA SHOULD BE WRFROM+256+66
7366  035262  052737  000200  003140          BIS     #OR,@#REINTO+4  ;SAVED RHCS2
7367  035270  042737  000100  003140          BIC     #IR,@#REINTO+4  ;SAVED RHCS2
7368  035276  052737  140000  003142          BIS     #SC!TRE,@#REINTO+6      ;SAVED RHCS1 SHOULD HAVE SC TRE
7369  035304  012737  001000  003144          MOV     #AOE,@#REINTO+10 ;SAVED RHER1 SHOULD HAVE AOE
7370  035312  017737  144320  003146          MOV     @RHDST,@#REINTO+12      ;SAVED RHDST SHOULD HAVE=
7371                                                                  ;RHDST IS UNDEFINED
7372  035320  012737  000633  003154          MOV     #411.,@#REINTO+20       ;SAVED DESIRED CYLINDER ADDRESS
7373  035326  013737  002004  003160          MOV     @#ATTENT,@#REINTO+24 ;SAVED RHAS SHOULD HAVE APPROPIATE BIT
7374  035334  052737  000001  003162          BIS     #DMD,@#REINTO+26        ;SAVED RHMR
7375  035342  052737  142000  003164          BIS     #ATA!ERR!LST,@#REINTO+30 ;SAVED RHDS1
7376
7377                                          ;AFTER A WRITE DATA COMMAND WITH AOE ERROR
7378                                          ;SAVE REGISTERS AGAIN SO THAT COMPARES CAN BE DONE
7379  035350  004037  041214                  JSR     R0,@#SAVER      ;SAVE
7380  035354  001624                          RH%C                    ;FROM
7381  035356  002070                          WRFROM                  ;TO
7382  035360  000021                          17.                     ;NUMBER OF REGISTERS SAVED
7383
7384                                          ;AS UPPER BYTE OF RHAS CAN BE CHANGING IN A DUAL PORT
7385                                          ;OPERATION THE UPPER BYTE OF RHAS WILL BE SAVED AS IS
7386                                          ;SO THAT THE COMPARES ARE ONLY VALID FOR THE LOWER BYTE
7387  035362  113737  003161  002115          MOVB    @#REINTO+25,@#WRFROM+25;SAVE UPPER RHAS
7388
7389
7390                                          ;COMPARE REGISTERS BEFORE WRITE DATA COMMAND
7391                                          ;WITH AFTER COMMAND
7392  035370  004037  041416                  JSR     R0,@#COMPAR     ;COMPARE
7393  035374  003134                          REINTO                  ;GOOD BUFFER
7394  035376  002070                          WRFROM                  ;TEST BUFFER
7395  035400  000021                          17.                     ;NUMBER OF REGISTERS
7396  035402  035410                          1$                      ;RETURN FOR ERROR
7397  035404  035410                          1$                      ;SAME
7398  035406  035430                          2$                      ;RETURN FOR GOOD COMPARISON
7399
7400  035410  013705  045272          1$.     MOV     @#ERWORD,R5     ;GETTING READY TO INDEX
7401  035414  060505                          ADD     R5,R5           ;DOUBLE ERROR WORD
7402  035416  016537  001622  040172          MOV     RH%C-2(R5),@#REGADR ;FAILING REG. ADDRESS
7403  035424  104001                          ERROR   1               ;FORCED AOE ERROR CAUSED IMPROPER
```

```
7404
7405  035426  000207                          RTS     PC              ;REGISTER CHANGE
7406                                                                  ;RETURN FOR FURTHER COMPARISONS
7407                                                                  ;NO ERRORS
7408  035430  005037  001774          2$:     CLR     @#ERFLG$        ;CLEAR ERROR FLAG
7409
7410                                  ;DATA IS TO BE CHECKED
7411  035434  004737  040126                  JSR     PC,@#PUTREG     ;SAVE REGISTERS
7412  035440  012700  000377                  MOV     #377,R0         ;GOOD DATA
7413  035444  012701  047070                  MOV     #DISK,R1        ;DATA WRITTEN INTO "DISK"
7414  035450  012702  000400                  MOV     #256.,R2        ;COUNTER
7415  035454  012737  000400  045272  3$:     MOV     #256.,@#ERWORD  ;FOR ERROR WORD
7416  035462  020021                          CMP     R0,(R1)+        ;COMPARE GOOD DATA WITH DATA ON DISK
7417  035464  001424                          BEQ     6$              ;BRANCH IF GOOD
7418  035466  010037  001124                  MOV     R0,@#$GDDAT     ;GOOD DATA
7419  035472  014137  001126                  MOV     -(R1),@#$BDDAT  ;BAD DATA
7420  035476  160237  045272                  SUB     R2,@#ERWORD     ;ERROR WORD NO
7421  035502  005737  001774                  TST     @#ERFLG$        ;ANY ERRORS ALREADY THERE?
7422  035506  001002                          BNE     4$              ;BRANCH IF YES
7423  035510  104004                          ERROR   4               ;ERROR ON WRITE DATA COMMAND WITH FORCED AOE
7424  035512  000401                          BR      5$              ;BRANCH TO AVOID PRINTING NEXT ERROR
7425  035514  104005                  4$:     ERROR   5               ;WORD NO GIVES WORD IN ERROR
7426  035516  005721                  5$:     TST     (R1)+           ;UNDO -(R1) FOR BAD DATA
7427  035520  013746  177570                  MOV     @#SWR,-(SP)     ;GET SWITCH SETTING
7428  035524  042716  177177                  BIC     #177177,(SP)    ;KEEP ONLY SWITCH 7 AND 8
7429  035530  022726  000200                  CMP     #SW07,(SP)+     ;IS 7 SET AND 8 RESET
7430
7431  035534  001402                          BEQ     7$              ;BRANCH OUT IF YES
7432  035536  005302                  6$:     DEC     R2              ;IF NOT COUNT 256 WORDS
7433  035540  001345                          BNE     3$              ;BRANCH IF 256. NOT DONE
7434
7435  035542                          7$:
```

```
7436
7437                                    ;*****************************************************************
7438                                    ;*TEST 114      MAKE CURRENT CYLINDER = 0
7439                                    ;*****************************************************************
7440  035542  000004                    TST114:*SCOPE
7441  035544  012706  001000                    MOV     #STACK,SP       ;RESET STACK
7442  035550  004737  040522                    JSR PC,@#CLDISK         ;INIT DRIVE
7443  035554  012777  000001  144070            MOV     #DMD,@RHMR      ;SET DIAGNOSTIC MODE
7444  035562  004037  043074                    JSR     R0,@#MAKECYL    ;SUBROUTINE TO GIVE A SEEK
7445                                                                    ;COMMAND FOLOWED BY A INIT
7446                                                                    ;THIS SHUOLD CHANGE RHCC
7447  035566  000000                            0                       ;CHANGE RHCC TO 0
7448
7449
7450
7451                                    ;*****************************************************************
7452                                    ;*TEST 115      ERROR REGISTER 1 - BIT #10 IAE
7453
7454                                    ;*      A READ HEADER AND DATA IS GIVEN TO TRACK 20, SECTOR 0
7455                                    ;*
7456                                    ;*      AN INDEX PULSE IS GIVEN TO GET RHLA TO 0
7457                                    ;*
7458                                    ;*      IAE BIT SHOULD SET
7459
7460                                    ;*****************************************************************
7461  035570  000004                    TST115: SCOPE
7462  035572  012706  001000                    MOV     #STACK,SP       ;RESET STACK
7463  035576  012737  000115  002020            MOV     #TTNO,@#TSTNM   ;THIS SAVES TEST NUMBER
7464  035604  004737  040522                    JSR     PC,@#CLDISK     ;CLEAR REGISTERS AND SET UNIT NO.
7465
7466                                    ;GIVE INDEX PULSE
7467  035610  012777  000001  144034            MOV     #DMD,@RHMR      ;SET DIAGNOSTIC MODE
7468  035616  052777  000004  144026            BIS     #MINX,@RHMR     ;SET INDEX
7469  035624  042777  000004  144020            BIC     #MINX,@RHMR     ;CLEAR INDEX
7470
7471
7472                                    ;THESE ARE REGULAR SETUPS
7473
7474  035632  012777  177400  143764            MOV     #-256.,@RHWC    ;256 DATA  WORDS 4 HEADER WORDS
7475  035640  012700  003134                    MOV     #REINTO,R0      ;THESE TWO INSTRUCTIONS GETS
7476  035644  010077  143756                    MOV     R0,@RHBA        ;ADDR. OF WRFROM INTO R0 AND
7477                                                                    ;BUS ADDRESS REGISTER
7478  035650  012720  010000                    MOV     #FMT22,(R0)+    ;FORMAT=16 BIT WORDS
7479                                                                    ;CYLINDER=0
7480  035654  012720  012000                    MOV     #12000,(R0)+    ;TRACK=20 SECTOR=0 KEYS=0
7481  035660  005020                            CLR     (R0)+           ;KEY1=0
7482  035662  005020                            CLR     (R0)+           ;KEY2=0
7483  035664  012705  000400                    MOV     #256.,R5        ;COUNTER
7484  035670  012720  177777            1$:     MOV     #-1,(R0)+       ;MOVE ALL ONES FOR DATA
7485  035674  005305                            DEC     R5
7486  035676  001374                            BNE     1$              ;BRANCH IF DATA NOT COMPLETE
7487  035700  012777  012000  143730            MOV     #12000,@RHDST   ;TRACK=20 SECTOR=0
7488
7489  035706  004737  040556                    JSR     PC,@#CHECKT     ;CHECK THAT DVA,RDY,DPR,DRY = 1
```

```
7490  035712  104400  057176                  TYPE    ,CPHALT         ;AND THAT NO OTHERS = 1.  CANNOT CON-
7491                                                                    ;TINUE TESTING IF BOTH AREN'T TRUE
7492  035716  000000                          HALT                    ;STOP THE TEST
7493
7494  035720  013711  002052                  MOV     @#REFOR,@R1     ;GET READY FOR WRITE HEADER AND
7495                                                                    ;DATA WITH 62 IN RHCS1
7496  035724  005037  001774                  CLR     @#ERFLG$        ;CLEAR ERROR FLAG
7497  035730  012777  010000  143704          MOV     #FMT22,@RHOF    ;FORMAT BIT=1 (16 BIT WORDS)
7498  035736  005077  143702                  CLR     @RHCA           ;CYLINDER =0
7499
7500                                          ;THE REGISTERS WILL BE SAVED IN REINTO BUFFER
7501  035742  004037  041214                  JSR     R0,@#SAVER      ;SAVE
7502  035746  001624                          RHMC                    ;FROM
7503  035750  003134                          REINTO                  ;TO
7504  035752  000023                          19.                     ;NUMBER SAVED
7505
7506                                          ;GO TO WRITE HEADER AND DATA
7507
7508  035754  013700  001652                  MOV     @#RHMR,R0       ;NOW R0 WAS MAINTENANCE REG. ADDR.
7509  035760  012710  000001                  MOV     #DMD,@R0        ;SET DIAGNOSTIC MODE
7510  035764  052777  000001  143640          BIS     #GO,@RHCS1      ;GO
7511
7512                                          ;CHANGE SAVED REGISTERS TO EXPECTED VALUE
7513  035772  052737  140000  003142          BIS     #SC!TRE,@#REINTO+6 ;SAVED RHCS1
7514  036000  012737  002000  003144          MOV     #IAE,@#REINTO+10 ;SAVED RHER1
7515  036006  012737  012001  003146          MOV     #12001,@#REINTO+12;SAVED RHDST
7516  036014  013737  002004  003160          MOV     @#ATTENT,@#REINTO+24 ;SAVED RHAS
7517  036022  052737  000001  003162          BIS     #DMD,@#REINTO+26 ;SAVED RHMR
7518  036030  052737  140000  003164          BIS     #ATA!ERR,@#REINTO+30 ;SAVED RHDS1
7519
7520                                          ;SAVE REGISTERS AGAIN SO THAT COMPARES CAN BE DONE
7521  036036  004037  041214                  JSR     R0,@#SAVER      ;SAVE
7522  036042  001624                          RHMC                    ;FROM
7523  036044  002070                          WRFROM                  ;TO
7524  036046  000023                          19.                     ;NUMBER OF REGISTERS SAVED
7525
7526                                          ;AS UPPER BYTE OF RHAS CAN BE CHANGING IN A DUAL PORT
7527                                          ;OPERATION THE UPPER BYTE OF RHAS WILL BE SAVED AS IS
7528                                          ;SO THAT THE COMPARES ARE ONLY VALID FOR THE LOWER BYTE
7529  036050  113737  003161  002115          MOVB    @#REINTO+25,@#WRFROM+25;SAVE UPPER RHAS
7530
7531
7532                                          ;COMPARE REGISTERS BEFORE READ IN PRESET COMMAND
7533                                          ;WITH AFTER COMMAND
7534  036056  004037  041416                  JSR     R0,@#COMPAR     ;COMPARE
7535  036062  003134                          REINTO                  ;GOOD BUFFER
7536  036064  002070                          WRFROM                  ;TEST BUFFER
7537  036066  000021                          17.                     ;NUMBER OF REGISTERS
7538  036070  036076                          2$                      ;RETURN FOR ERROR
7539  036072  036076                          2$                      ;SAME
7540  036074  036116                          3$                      ;RETURN FOR GOOD COMPARISON
7541
7542  036076  013705  045272          2$:     MOV     @#ERWORD,R5     ;GETTING READY TO INDEX
7543  036102  060505                          ADD     R5,R5           ;DOUBLE ERROR WORD
```

```
7544  036104  016537  001622  040172          MOV     RH##C-2(R5),@#REGADR ;FAILING REG. ADDRESS
7545  036112  104001                          ERROR   1                    ;FORCED IAE CAUSED IMPROPER
7546                                                                       ;REGISTER CHANGE
7547  036114  000207                          RTS     PC                   ;RETURN FOR FURTHER COMPARISONS
7548
7549                                                                       ;NO ERRORS
7550
7551  036116  004737  040522          3$:     JSR     PC,@#CLDISK          ;CLEAR GO BIT
7552
```

```
                                ;*****************************************************************
                                ;*TEST 116      ERROR REGISTER 1- BIT #10 IAE
                                ;
                                ;*      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
                                ;
                                ;*****************************************************************
036122  000004                  TST116: SCOPE
036124  012706  001000                  MOV     #STACK,SP       ;RESET STACK
036130  012737  000116  002020          MOV     #TTNO,@#TSTNM   ;THIS SAVES TEST NUMBER
036136  004737  040522                  JSR     PC,@#CLDISK     ;CLEAR REGISTERS AND SET UNIT NO.
                                        ;GIVE INDEX PULSE
036142  012777  000001  143502          MOV     #DMD,@RHMR      ;SET DIAGNOSTIC MODE
036150  052777  000004  143474          BIS     #MINX,@RHMR     ;SET INDEX
036156  042777  000004  143466          BIC     #MINX,@RHMR     ;CLEAR INDEX
                                        ;THESE ARE REGULAR SETUPS
036164  012777  177400  143432          MOV     #-256.,@RHWC    ;256 DATA WORDS 4 HEADER WORDS
036172  012700  002070                  MOV     #WRFROM,R0      ;THESE TWO INSTRUCTIONS GETS
036176  010077  143424                  MOV     R0,@RHBA        ;ADDR. OF WRFROM INTO R0 AND
                                                                ;BUS ADDRESS REGISTER
036202  012720  010000                  MOV     #FMT22,(R0)+    ;FORMAT=16 BIT WORDS
                                                                ;CYLINDER=0
036206  012720  000026                  MOV     #22.,(R0)+      ;TRACK=0, SECTOR=22, KEYS=0
036212  005020                          CLR     (R0)+           ;KEY1=0
036214  005020                          CLR     (R0)+           ;KEY2=0
036216  012705  000400                  MOV     #256.,R5        ;COUNTER
036222  012720  177777          1$:     MOV     #-1,(R0)+       ;MOVE ALL ONES FRO DATA
036226  005305                          DEC     R5
036230  001374                          BNE     1$              ;BRANCH IF DATA NOT COMPLETE
036232  012777  000026  143376          MOV     #22.,@RHDST     ;TRACK=0 SECTOR=22
036240  004737  040556                  JSR     PC,@#CHECKT     ;CHECK THAT DVA,RDY,DPR,DRY = 1
036244  104400  057176                  TYPE    ,CPHALT         ;AND THAT NO OTHERS = 1.  CANNOT CON-
                                                                ;TINUE TESTING IF BOTH AREN'T TRUE
036250  000000                          HALT                    ;STOP THE TEST
036252  013711  002046                  MOV     @#WRIFOR,@R1    ;GET READY FOR WRITE HEADER AND
                                                                ;DATA WITH 62 IN RHCS1
036256  005037  001774                  CLR     @#ERFLG$        ;CLEAR ERROR FLAG
036262  012777  010000  143352          MOV     #FMT22,@RHOF    ;FORMA BIT=1 (16 BIT WORDS)
036270  005077  143350                  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
036274  004037  041214                  JSR     R0,@#SAVER      ;SAVE
036300  001634                          RHER1                   ;FROM
036302  003134                          REINTO                  ;TO
036304  000015                          13.                     ;NUMBER SAVED

                                        ;GO TO WRITE HEADER AND DATA

036306  013700  001652                  MOV     @#RHMR,R0       ;NOW R0 HAS MAINTENANCE REG. ADDR.
036312  012710  000001                  MOV     #DMD,@R0        ;SET DIAGNOSTIC MODE
036316  052777  000001  143306          BIS     #GO,@RHCS1      ;GO

                                        ;CHANGE SAVED REGISTERS TO EXPECTED VALUE
036324  012737  002000  003134          MOV     #IAE,@#REINTO   ;SAVED RHER1
036332  012737  000027  003136          MOV     #23.,@#REINTO+2 ;SAVED RHDST
036340  013737  002004  003150          MOV     @#ATTENT,@#REINTO+14 ;SAVED RHAS
036346  052737  000001  003152          BIS     #DMD,@#REINTO+16 ;SAVED RHMR
036354  052737  140000  003154          BIS     #ATAERR,@#REINTO+20 ;SAVED RHDS1

                                        ;SAVE REGISTERS AGAIN SO THAT COMPARES CAN BE DONE
036362  004037  041214                  JSR     R0,@#SAVER      ;SAVE
036366  001634                          RHER1                   ;FROM
036370  002070                          WRFROM                  ;TO
036372  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
036374  113737  003151  002105          MOVB    @#REINTO+15,@#WRFROM+15;SAVE UPPER RHAS


                                        ;COMPARE REGISTERS BEFORE READ IN PRESET COMMAND
                                        ;WITH AFTER COMMAND
036402  004037  041416                  JSR     R0,@#COMPAR     ;COMPARE
036406  003134                          REINTO                  ;GOOD BUFFER
036410  002070                          WRFROM                  ;TEST BUFFER
036412  000015                          13.                     ;NUMBER OF REGISTERS
036414  036422                          2$                      ;RETURN FOR ERROR
036416  036422                          2$                      ;SAME
036420  036442                          3$                      ;RETURN FOR GOOD COMPARISON

036422  013705  045272          2$:     MOV     @#ERWORD,R5     ;GETTING READY TO INDEX
036426  060505                          ADD     R5,R5           ;DOUBLE ERROR WORD
036430  016537  001632  040172          MOV     RHER1-2(R5),@#REGADR ;FAILING REG. ADDRESS
036436  104001                          ERROR   1               ;FORCED IAE CAUSED IMPROPER
                                                                ;REGISTER CHANGE
036440  000207                          RTS     PC              ;RETURN FOR FURTHER COMPARISONS

                                        ;NO ERRORS
```

```
MAINDEC-11-DZRPS-C, RJP04 DISKLESS RH11 TEST-PART 1     MACY11 27(663)  8-OCT-75  16:14  PAGE 182                    SEQ 0224
DZRPSD.P11      T116    ERROR REGISTER 1- BIT #10 IAE

  7661  036442  004737  040522          35:     JSR     PC,@#CLDISK     ;CLEAR GO BIT
  7662
  7663
```

```
                                ;*****************************************************************
                                ;*TEST 117      ERROR REGISTER 1- BIT #10 IAE
                                ;
                                ;*      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
                                ;
                                ;*****************************************************************
036446  000004                  TST117: SCOPE
036450  012706  001000                  MOV     #STACK,SP       ;RESET STACK
036454  012737  000117  002020          MOV     #TTNU,@#TSTNM   ;THIS SAVES TEST NUMBER
036462  004737  040522                  JSR     PC,@#CLDISK     ;CLEAR REGISTERS AND SET UNIT NO.
                                
                                        ;GIVE INDEX PULSE
036466  012777  000001  143156          MOV     #DMD,@RHMR      ;SET DIAGNOSTIC MODE
036474  052777  000004  143150          BIS     #MINX,@RHMR     ;SET INDEX
036502  042777  000004  143142          BIC     #MINX,@RHMR     ;CLEAR INDEX
                                
                                
                                        ;THESE ARE REGULAR SETUPS
                                
036510  012777  177400  143106          MOV     #-256.,@RHWC    ;256 DATA WORDS & HEADER WORDS
036516  012700  002070                  MOV     #WRFROM,R0      ;THESE TWO INSTRUCTIONS GETS
036522  010077  143100                  MOV     R0,@RHBA        ;ADDR. OF WRFROM INTO R0 AND
                                                                ;BUS ADDRESS REGISTER
036526  012705  000400                  MOV     #256.,R5        ;COUNTER
036532  012720  177777          1$:     MOV     #-1,(R0)+       ;MOVE ALL ONES FRO DATA
036536  005305                  DEC     R5
036540  001374                  BNE     1$              ;BRANCH IF DATA NOT COMPLETE
036542  012777  000000  143066          MOV     #0.,@RHDST      ;TRACK=0 SECTOR=0
                                
036550  004737  040556                  JSR     PC,@#CHECKT     ;CHECK THAT DVA,RDY,DPR,DRY = 1
036554  104400  057176                  TYPE    ,CPHALT         ;AND THAT NO OTHERS = 1.  CANNOT CON-
                                                                ;TINUE TESTING IF BOTH AREN'T TRUE
036560  000000                  HALT                            ;STOP THE TEST
                                
036562  013711  002044                  MOV     @#WRIDAT,@R1    ;GET READY FOR WRITE
                                                                ;DATA WITH 60 IN RHCS1
036566  005037  001774                  CLR     @#ERFLG$        ;CLEAR ERROR FLAG
036572  012777  010000  143042          MOV     #FMT22,@RHOF    ;FORMA BIT=1 (16 BIT WORDS)
036600  012777  000633  143036          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
```

```
7718  036606  004037  041214                  JSR     R0,@#SAVER      ;SAVE
7719  036612  001634                          RHER1                   ;FROM
7720  036614  003134                          REINTO                  ;TO
7721  036616  000015                          13.                     ;NUMBER SAVED
7722
7723                                          ;GO TO WRITE HEADER AND DATA
7724
7725  036620  013700  001652                  MOV     @#RHMR,R0       ;NOW R0 HAS MAINTENANCE REG. ADDR.
7726  036624  012710  000001                  MOV     #DMD,@R0        ;SET DIAGNOSTIC MODE
7727  036630  052777  000001  142774          BIS     #GO,@RHCS1      ;GO
7728
7729                                          ;CHANGE SAVED REGISTERS TO EXPECTED VALUE
7730  036636  012737  002000  003134          MOV     #IAE,@#REINTO   ;SAVED RHER1
7731  036644  012737  000001  003136          MOV     #1.,@#REINTO+2;SAVED RHDST
7732  036652  013737  002004  003150          MOV     @#ATTENT,@#REINTO+14 ;SAVED RHAS
7733  036660  052737  000001  003152          BIS     #DMD,@#REINTO+16 ;SAVED RHMR
7734  036666  052737  140000  003154          BIS     #ATA!ERR,@#REINTO+20 ;SAVED RHDS1
7735
7736                                          ;SAVE REGISTERS AGAIN SO THAT COMPARES CAN BE DONE
7737  036674  004037  041214                  JSR     R0,@#SAVER      ;SAVE
7738  036700  001634                          RHER1                   ;FROM
7739  036702  002070                          WRFROM                  ;TO
7740  036704  000015                          13.                     ;NUMBER OF REGISTERS SAVED
7741
7742                                          ;AS UPPER BYTE OF RHAS CAN BE CHANGING IN A DUAL PORT
7743                                          ;OPERATION THE UPPER BYTE OF RHAS WILL BE SAVED AS IS
7744                                          ;SO THAT THE COMPARES ARE ONLY VALID FOR THE LOWER BYTE
7745  036706  113737  003151  002105          MOVB    @#REINTO+15,@#WRFROM+15;SAVE UPPER RHAS
7746
7747
7748                                          ;COMPARE REGISTERS BEFORE READ IN PRESET COMMAND
7749                                          ;WITH AFTER COMMAND
7750  036714  004037  041416                  JSR     R0,@#COMPAR     ;COMPARE
7751  036720  003134                          REINTO                  ;GOOD BUFFER
7752  036722  002070                          WRFROM                  ;TEST BUFFER
7753  036724  000015                          13.                     ;NUMBER OF REGISTERS
7754  036726  036734                          2$                      ;RETURN FOR ERROR
7755  036730  036734                          2$                      ;SAME
7756  036732  036754                          3$                      ;RETURN FOR GOOD COMPARISON
7757
7758  036734  013705  045272          2$:     MOV     @#ERWORD,R5     ;GETTING READY TO INDEX
7759  036740  060505                          ADD     R5,R5           ;DOUBLE ERROR WORD
7760  036742  016537  001632  040172          MOV     RHER1-2(R5),@#REGADR ;FAILING REG. ADDRESS
7761  036750  104001                          ERROR   1               ;FORCED IAE CAUSED IMPROPER
7762                                                                  ;REGISTER CHANGE
7763  036752  000207                          RTS     PC              ;RETURN FOR FURTHER COMPARISONS
7764
7765                                                                  ;NO ERRORS
7766
7767  036754  004737  040522          3$:     JSR     PC,@#CLDISK     ;CLEAR GO BIT
```

```
                                ;*****************************************************************
                                ;*TEST 120      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
                                ;
                                ;*****************************************************************
036760  000004                  TST120: SCOPE
036762  012767  000001  142214          MOV     #1,$TIMES       ;;DO 1 ITERATION
036770  012767  000000  141000          MOV     #0,PS           ;REINSTATE PS TO 0
036776  104400  037004                  TYPE    ,.+4            ;;TYPE ASCIZ STRING
037002  000425                          BR      64$             ;;GET OVER THE ASCIZ
                                        ;;.ASCIZ      <15><12>/TOTAL ERRORS ON THIS PASS ON  UNIT NO. /
037056                          64$:
037056  013746  001762                  MOV     @#UNIT,-(SP)    ;GET READY TO TYPE UNIT NUMBER
037062  104410                          TYPOS
037064  104400  037072                  TYPE    ,.+4            ;;TYPE ASCIZ STRING
037070  000402                          BR      65$             ;;GET OVER THE ASCIZ
                                        ;;.ASCIZ      /= /
037076                          65$:
037076  013746  001112                  MOV     @#SERTTL,-(SP)  ;GET READY TO TYPE NUMBER OF ERRORS
037102  104410                          TYPOS
037104  005037  001112                  CLR     @#SERTTL        ;CLEAR TOTAL NUMBER OF ERRORS
0371..  005737  001770                  TST     @#SELECT        ;STARTING FROM 200 ?
0371..  001415                          BEQ     3$              ;BRANCH IF YES
037116  005067  141760                  CLR     $TSTNM          ;CLEAR TEST NUMBER
037122  005237  001100                  INC     @#$PPSS         ;INCREASE PASS COUNT
037126  104400  037324                  TYPE    ,SENUMG         ;TYPE END PASS #
037132  013746  001100                  MOV     @#$PASS,-(SP)
037136  104410                          TYPOS
037140  104400  037341                  TYPE    ,SENULL
037144  000137  007272                  JMP     @#TST5          ;JUMP TEST 5
037150  005337  001764          3$:     DEC     @#NOUNITS       ;NO. OF UNITS PRESENT DECREMENT
037154  001413                          BEQ     $EOP            ;BRANCH IF ALL DRIVES COMPLETE
037156  013700  001762                  MOV     @#UNIT,R0       ;UNIT UNDER TEST
037162  012701  001742                  MOV     #UNITS,R1       ;TABLE
037166  022100                  1$:     CMP     (R1)+,R0        ;IS THIS UNIT JUST TESTED
037170  001401                          BEQ     2$              ;BRANCH IF YES
037172  000775                          BR      1$              ;BRANCH IF NO
037174  011137  001762          2$:     MOV     (R1),@#UNIT     ;THIS IS NEXT UNIT
037200  000137  007272                  JMP     @#TST5          ;GO FOR NEXT TESTS.

                                        .SBTTL
                                        .SBTTL  ***SUBROUTINES***
                                        .SBTTL

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

                                .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
                                        
037204                                  $EOP:
037204  000004                                SCOPE
037206  005067  141670                      CLR     $TSTNM          ;;ZERO THE TEST NUMBER
037212  005067  141766                      CLR     $TIMES          ;;ZERO THE NUMBER OF ITERATIONS
037216  005267  141656                      INC     $PASS           ;;INCREMENT THE PASS NUMBER
037222  042767  100000  141650            BIC     #100000,$PASS   ;;DON'T ALLOW A NEG. NUMBER
037230  005327                          $EOPCT: DEC     (PC)+           ;;LOOP?
037232  000001                                  .WORD   1
037234  003031                                  BGT     $DOAGN          ;;YES
037236  012737                                  MOV     (PC)+,@(PC)+    ;;RESTORE COUNTER
037240  000001                          $ENDCT: .WORD   1
037242  037232                                  $EOPCT
037244  104400  037324                        TYPE    ,$ENDMG         ;;TYPE "END PASS #"
037250  016746  141624                        MOV     $PASS,-(SP)     ;;SAVE $PASS FOR TYPEOUT
037254  104410                              TYPDS                   ;;GO TYPE--DECIMAL ASCII WITH SIGN
037256  104400  037341                        TYPE    ,$ENULL         ;;TYPE A NULL CHARACTER
037262  013700  000042                  $GET42: MOV     @#42,R0         ;;GET MONITOR ADDRESS
037266  001414                              BEQ     $DOAGN          ;;BRANCH IF NO MONITOR
037270  022700  037310                      CMP     #$ENDAD,R0      ;;IS MONITOR ACT11?
037274  001004                              BNE     $RESET          ;;NO--BRANCH (IT'S XXDP)
037276  022760  177777  000002              CMP     #-1,2(R0)       ;;YES--IS THIS THE LAST PASS?
037304  001001                              BNE     $ENDAD          ;;NO--MAKE ANOTHER PASS
037306  000005                          $RESET: RESET                   ;;CLEAR THE WORLD
037310  004710                          $ENDAD: JSR     PC,(R0)         ;;GO TO MONITOR
037312  000240                                  NOP                     ;;SAVE ROOM
037314  000240                                  NOP                     ;;FOR
037316  000240                                  NOP                     ;;ACT11
037320  000137  005346                  $DOAGN: JMP     @#TST1          ;;RETURN
037324  005015  047105  020104          $ENDMG: .ASCIZ  <15><12>/END PASS #/
037332  040520  051523  021440
037340     000
037341     377      377      000      $ENULL: .BYTE   -1,-1,0         ;;NULL CHARACTER STRING
                                        
                                        
                                        ;*****************************************************************
                                        ;*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.
                                        ;*****************************************************************


037344  000000                  TESTAD: 0                       ;FIRST ADDRESS OF TEST

037346                          OPERSEL:
037346  005067  140424                  CLR     PS              ;MAKE PROCESSOR STATUS ZERO
037352  104400  037360                  TYPE    ,.+4            ;;TYPE ASCIZ STRING
037356  000421                          BR      64$             ;;GET OVER THE ASCIZ
                                        ;;.ASCIZ        <15><12>/THE PROGRAM WAS IN TEST NUMBER /
037422                          64$:
037422  013746  002020                  MOV     @#TSTNM,-(SP)   ;GET READY TO TYPE TEST
037426  104402                          TYPOC                   ;NUMBER
037430  104400  037436                  TYPE    ,.+4            ;;TYPE ASCIZ STRING
037434  000414                          BR      65$             ;;GET OVER THE ASCIZ
                                        ;;.ASCIZ        <15><12>/THE LOOP BACK PC WAS /
037466                          65$:
037466  013746  001110                  MOV     @#$LPERR,-(SP)  ;GET READY TO TYPE LOOP BACK PC
037472  104402                          TYPOC
037474  104400  001215                  TYPE    ,$CRLF
037500  104400  037506                  TYPE    ,.+4            ;;TYPE ASCIZ STRING
037504  000430                          BR      66$             ;;GET OVER THE ASCIZ
                                        ;;.ASCIZ        <15><12>/SET SWITCH FOR LOOP ON ERROR OR LOOP ON TEST/
037566                          66$:
037566  104400  037574                  TYPE    ,.+4            ;;TYPE ASCIZ STRING
037572  000430                          BR      67$             ;;GET OVER THE ASCIZ
                                        ;;.ASCIZ        <15><12>/TYPE THE FIRST PC OF THE TEST TO BE LOOPED ON/
037654                          67$:
037654  104400  037662                  TYPE    ,.+4            ;;TYPE ASCIZ STRING
037660  000422                          BR      68$             ;;GET OVER THE ASCIZ
                                        ;;.ASCIZ        <15><12>/ FOLLOWED BY A CARRIAGE RETURN /<15><12>
037726                          68$:
037726  104416                          RDOCT
037730  062716  000002                  ADD     #2,(SP)         ;GET LPADR
037734  012637  001106                  MOV     (SP)+,@#$LPADR
037740  104400  037746                  TYPE    ,.+4            ;;TYPE ASCIZ STRING
037744  000417                          BR      69$             ;;GET OVER THE ASCIZ
                                        ;;.ASCIZ        <15><.2>/TYPE THE PC WHERE YOU WANT/
040004                          69$:
040004  104400  040012                  TYPE    ,.+4            ;;TYPE ASCIZ STRING
040010  000440                          BR      70$             ;;GET OVER THE ASCIZ
                                        ;;.ASCIZ        <15><12>/ THE PROGRAM TO LOOP BACK TO FOLLOWED BY A CARRIAGE RETUR
040112                          70$:
040112  104416                          RDOCT
040114  012637  001110                  MOV     (SP)+,@#$LPERR  ;GET LPERR
040120  013746  001106                  MOV     @#$LPADR,-(SP)
040124  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

040126                          PUTREG:
040126  010046                          MOV     R0,-(SP)        ;;PUSH R0 ON STACK
040130  010146                          MOV     R1,-(SP)        ;;PUSH R1 ON STACK
040132  010246                          MOV     R2,-(SP)        ;;PUSH R2 ON STACK
040134  012700  001624                  MOV     #RHWC,R0        ;STARTING ADDRESS OF REG
040140  012701  001674                  MOV     #WC,R1          ;STARTING ADDRESS OF WERE SAVED
040144  012702  000023                  MOV     #RHCC-RHWC+2/2,R2 ;NUMBER OF REG. INTO R2
040150  013021                  10$:     MOV     @(R0)+,(R1)+    ;SAVE HARDWARE REG.
040152  005302                          DEC     R2
040154  001375                          BNE     10$
040156  012602                          MOV     (SP)+,R2        ;;POP STACK INTO R2
040160  012601                          MOV     (SP)+,R1        ;;POP STACK INTO R1
040162  012600                          MOV     (SP)+,R0        ;;POP STACK INTO R0
040164  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

040166  000000                  MASK:   0                       ;BITS UNDER TEST
040170  000000                  LERR:   0                       ;ERROR HLT ADDRESS
040172  000000                  REGADR: 0

040174  012567  177766          BITST:  MOV     (R5)+,  MASK    ;FETCH DATA MASK
040200  012504                          MOV     (R5)+,  R4      ;GET ADDRESS OF REG. UNDER TEST
040202  010467  177764                  MOV     R4,     REGADR
040206  010567  177756                  MOV     R5,     LERR    ;GET ERROR RETURN ADDR.
040212  062705  000004                  ADD     #4,     R5      ;MODIFY RETURN ADDR. TO JUMP OVER RTS
040216  012703  000001                  MOV     #1,     R3      ;INITIALIZE DATA PATTERN
040222  004767  000016          BLT1:   JSR     PC,     BLT2    ;OUTPUT FLOATING ZERO
040226  004767  000012                  JSR     PC,     BLT2    ;OUTPUT FLOATING ONE
040232  000241                          CLC
040234  006103                          ROL     R3              ;SHIFT PATTERN
040236  005703                          TST     R3
040240  001370                          BNE     BLT1            ;BRANCH IF NOT COMPLETE
040242  000205                          RTS     R5              ;RETURN TO TEST
040244  005103                  BLT2:   COM     R3              ;COMPLEMENT PATTERN
```

```
 7984  040246  012737  040254  040502          MOV     #BLT3,  @#LAD   ;SET SCOPE LOOP
 7985  040254  032737  001000  177570  BLT3:   BIT     #SW09,@#SWR     ;LOOP ON ERROR
 7986  040262  001411                          BEQ     4$              ;BRANCH IF NO
 7987  040264  105737  001103                  TSTB    @#ERFLG         ;ANY ERRORS
 7988  040270  001406                          BEQ     4$      ;BRANCH IF NO
 7989  040272  000005                          RESET                   ;START WITH AN INIT
 7990  040274  013777  001762  141326          MOV     @#UNIT,@RHCS2   ;SET UNIT NUMBER UNDER TEST
 7991  040302  004737  052512                  JSR     PC,@#TKINT      ;INITILIZE TK
 7992                                                                  ;INIT FOR SCOPING LOOPS
 7993  040306  010337  001124          4$:     MOV     R3,@#GDDAT      ;STORE GOOD DATA
 7994  040312  005137  040166                  COM     @#MASK          ;AND MASK WITH PATTERN
 7995  040316  043737  040166  001124          BIC     @#MASK, @#GDDAT         ;CLEAR THE REST
 7996  040324  005137  040166                  COM     @#MASK          ;RESTORE MASK
 7997  040330  013714  001124                  MOV     @#GDDAT,(R4)    ;OUTPUT TO REGISTER
 7998  040334  011437  001126                  MOV     (R4),@#BDDAT    ;INPUT FROM REGISTER
 7999  040340  005137  040166                  COM     @#MASK
 8000  040344  043737  040166  001126          BIC     @#MASK,@#BDDAT ;AND MASK OUT RECEIVED DATA
 8001  040352  005137  040166                  COM     @#MASK          ;RESTORE MASK
 8002  040356  023737  001124  001126          CMP     @#GDDAT,@#BDDAT         ;IS DATA CORRECT
 8003  040364  001424                          BEQ     1$              ;BRANCH IF GOOD
 8004  040366  011437  001126                  MOV     (R4),@#BDDAT
 8005  040372  023704  001632                  CMP     @#RHCS1,R4      ;REGISTER UNDER TEST RHCS1?
 8006  040376  001004                          BNE     2$              ;BRANCH IF NOT
 8007  040400  052737  004200  001124          BIS     #RDY!DVA,@#GDDAT;SET RDY AND DVA
 8008  040406  000410                          BR      3$
 8009  040410  023704  001630          2$:     CMP     @#RHCS2,R4      ;REGISTER UNDER TEST RHCS2?
 8010  040414  001005                          BNE     3$              ;BRANCH IF NOT
 8011  040416  011446                          MOV     @R4,-(SP)       ;GET RHCS2
 8012  040420  042716  177477                  BIC     #^C<IR!OR>,(SP) ;KEEP IR AND OR BIT
 8013  040424  052637  001124                  BIS     (SP)+,@#GDDAT   ;SET IR OR BITS IF NEEDED
 8014  040430  004777  177534          3$:     JSR     PC,     @LERR   ;GO TO REPORT ERROR
 8015  040434  000240                          NOP                     ;REPLACE BY 104420 FOR LOCAL SCOPE LOOP
 8016  040436  000207                  1$:     RTS     PC
 8017
 8018
 8019                                          .SBTTL  CLEAR MEMORY ROUTINE
 8020
 8021                                  ;       THIS CLEARS ANY BLOCK OF MEMORY
 8022                                  ;       FILLING IT WITH ANY DATA
 8023                                  ;
 8024                                  ;       CALL
 8025                                  ;       JSR     R0,CLAREA
 8026                                  ;       X                       ;STARTING ADDRESS OF BLOCK
 8027                                  ;       Y
 8028                                  ;       Z                       ;DATA TO BE FILLED
 8029                                  ;R1 WILL HAVE STARTING ADDRESS OF BLOCK TO BE FILLED
 8030                                  ;R2 AFTER SUBTRACTION WILL HAVE TWICE NUMBER OF LOCATIONS
 8031                                  ;R3 WILL HAVE DATA TO BE FILLED
 8032                                  ;TO AVOID DIVIDE ROUTINE TWO DECREMENT R2 WILL BE USED
 8033
 8034
 8035  040440                          CLAREA:
 8036  040440  010146                          MOV     R1,-(SP)        ;;PUSH R1 ON STACK
 8037  040442  010246                          MOV     R2,-(SP)        ;;PUSH R2 ON STACK
```

N02


```
 8038  040444  010346                        MOV     R3,-(SP)        ;;PUSH R3 ON STACK
 8039  040446  012001                        MOV     (R0)+,R1        ;FROM
 8040  040450  012002                        MOV     (R0)+,R2        ;TO
 8041  040452  012003                        MOV     (R0)+,R3        ;DATA
 8042  040454  160102                        SUB     R1,R2           ;NO. OF LOCATIONS MINUS TWO
 8043  040456  062702  000002                ADD     #2,R2           ;GET TWICE NO OF LOCATIONS
 8044  040462  010321              1$:       MOV     R3,(R1)+        ;MOVE IN DATA
 8045  040464  005302                        DEC     R2
 8046  040466  005302                        DEC     R2
 8047  040470  001374                        BNE     1$              ;BRANCH IF NOT COMPLETE
 8048  040472  012603                        MOV     (SP)+,R3        ;;POP STACK INTO R3
 8049  040474  012602                        MOV     (SP)+,R2        ;;POP STACK INTO R2
 8050  040476  012601                        MOV     (SP)+,R1        ;;POP STACK INTO R1
 8051  040500  000200                        RTS     R0              ;RETURN
```

```
8052                                                    .SBTTL  LOCAL TRAPS
8053  040502  000000                            LAD:    0
8054
8055  040504  032737  001000  177570  T.SCOP:   BIT     #SW09,  @#SWR
8056  040512  001402                            BEQ     1$
8057  040514  013716  040502                    MOV     @#LAD,  (SP)
8058  040520  000002                    1$:     RTI
8059
8060                                            ;EXAMPLE OF THE USE OF THE ABOVE
8061                                            ;THIS WILL LOOP BETWEEN X: AND SCOP1 PROVIDED THERE IS NO "NEWTST"
8062                                            ;MOV    #X,     @#LAD
8063                                            ;X:     ---     ---
8064                                            ;       ---     ---
8065                                            ;       ---     ---
8066                                            ;       SCOP1
8067
8068
8069                                                    .SBTTL  CLEAR DISK ROUTINE
8070
8071  040522  013701  001632            CLDISK: MOV     @#RHCS1,        R1      ;R1 WILL BE CONTROL AND STATUS1
8072  040526  013702  001630                    MOV     @#RHCS2,        R2      ;R2 WILL BE CONTROL AND STATUS2
8073  040532  013703  001654                    MOV     @#RHDS1,        R3      ;R3 WILL BE DISK STATUS REGISTER1
8074  040536  013704  001634                    MOV     @#RHER1,        R4      ;R4 WILL BE ERROR REGISTER #1
8075
8076  040542  012712  000040                    MOV     #CLR,@R2                ;CLEAR ALL REG.
8077  040546  013712  001762                    MOV     @#UNIT,@R2      ;REINSTATE UNIT NO.
8078  040552  005011                            CLR     @R1                     ;CLEAR FUNCTION BITS
8079  040554  000207                            RTS     PC
```

```
8080
8081
8082                                        .SBTTL  CHECK DISK STATUS ROUTINES
8083
8084
8085                                ;THIS CHECKS THAT DEVICE AVAILABLE (DVA) AND READY (RDY) IN RHCS1 = 1
8086                                ;AND CHECKS THAT DEVICE PRESENT (DPR), DEVICE READY (DRY) IN RHDS1 = 1
8087                                ;IT ALSO CHECKS THAT NO OTHER BITS IN THESE REGISTERS = 1
8088
8089  040556  011637  002002        CHECKT: MOV     (SP),@#PCJSR    ;SAVE PC OF JSR+4
8090  040562  162737  000004  002002        SUB     #4,@#PCJSR      ;GET PC OF JSR
8091  040570  004737  040126              JSR     PC,@#PUTREG     ;SAVE REGISTERS
8092  040574  022737  004200  001702        CMP     #DVA!RDY,@#CS1  ;RHCS1 SHOULD HAVE DEVICE AVAILABLE
8093                                                              ;AND BE READY
8094  040602  001423                       BEQ     3$              ;BRANCH IF GOOD
8095
8096  040604  032737  004000  001702        BIT     #DVA,@#CS1      ;BAD SO TEST DEVICE AVAILABLE
8097  040612  001004                       BNE     1$              ;BRANCH IF DVA THERE
8098  040614  010137  001122              MOV     R1,@#$BDADR     ;ADDRESS OF BAD REGISTER (RHCS1)
8099  040620  104026                       ERROR   26              ;RHCS1 DID NOT HAVE DEVICE
8100                                                              ;AVAILABLE AT START OF TEST
8101  040622  000413                       BR      3$              ;BRANCH TO NEXT COMPARE
8102  040624  032737  000200  001702  1$:   BIT     #RDY,@#CS1      ;TEST READY
8103  040632  001003                       BNE     2$              ;IF RDY THERE BRANCH
8104  040634  010137  001122              MOV     R1,@#$BDADR     ;ADDRESS OF BAD REGISTER (RHCS1)
8105  040640  104026                       ERROR   26              ;RHCS1 DID NOT HAVE READY
8106                                                              ;RIGHT AT START OF TEST
8107  040642  000403                 2$:   BR      3$              ;BRANCH TO NEXT COMPARE
8108  040644  010137  001122              MOV     R1,@#$BDADR     ;ADDRESS OF BAD REGISTER (RHCS1)
8109  040650  104026                       ERROR   26              ;RHCS1 HAD SOME BITS OTHER
8110                                                              ;THAN DVA AND ROY SET
8111                                                              ;ALL OTHER BITS SHOULD BE 0
8112
8113  040652  013746  001724         3$:   MOV     @#DS1,-(SP)     ;GET RHDS1
8114  040656  042716  001100              BIC     #VV!PROG,(SP)   ;CLEAR VV AND PROGRAMABLE BIT
8115  040662  022726  000600              CMP     #DPR!DRY,(SP)+;RHDS1 SHOULD HAVE THESE SET
8116  040666  001424                       BEQ     8$              ;BRANCH IF THEY ARE
8117
8118  040670  032737  000400  001724  4$:   BIT     #DPR,@#DS1      ;TEST DRIVE PRESENT
8119  040676  001004                       BNE     5$              ;CONTINUE IF THERE
8120  040700  010337  001122              MOV     R3,@#$BDADR     ;ADDRESS OF BAD REGISTER (RHDS1)
8121  040704  104026                       ERROR   26              ;RHDS1 DOES NOT HAVE DPR
8122  040706  000413                       BR      7$              ;BRANCH OUT
8123  040710  032737  000200  001724  5$:   BIT     #DRY,@#DS1      ;TEST DRIVE READY
8124  040716  001004                       BNE     6$              ;IF DPR WAS THERE, BRANCH IF GOOD
8125  040720  010337  001122              MOV     R3,@#$BDADR     ;ADDRESS OF BAD REGISTER (RHDS1)
8126  040724  104026                       ERROR   26              ;RHDS1 DOES NOT HAVE DRY
8127  040726  000403                       BR      7$              ;BRANCH OUT
8128  040730  010337  001122         6$:   MOV     R3,@#$BDADR     ;ADDRESS OF BAD REGISTER (RHDS1)
8129  040734  104026                       ERROR   26              ;RHDS1 HAS SOME BITS OTHER
8130                                                              ;THAN MOL, DRY, DPR, SET
8131                                                              ;ALL OTHER BITS SHOULD BE 0
8132  040736  000207                 7$:   RTS     PC              ;RETURN TO TEST AND HALT
8133
```

```
 8134  040740  062716  000006          8$:     ADD     #6,(SP)         ;ADJUST STACK TO GET OVER HALT IN TEST
 8135  040744  000207                          RTS     PC              ;RETURN TO TEST AND CONTINUE TESTING
 8136
```

```
8137
8138                                    ;THIS CHECKS THAT DEVICE AVAILABLE (DVA) AND READY (RDY) IN RHCS1 = 1
8139                                    ;AND CHECKS THAT DEVICE PRESENT (DPR), DEVICE READY (DRY) IN RHDS1 = 1
8140
8141  040746  011637  002002          CHECKE: MOV     (SP),@#PCJSR    ;SAVE PC OF JSR+4
8142  040752  162737  000004  002002          SUB     #4,@#PCJSR      ;GET PC OF JSR
8143  040760  004737  040126                  JSR     PC,@#PUTREG     ;READ & SAVE REGISTERS
8144  040764  032737  000200  001702          BIT     #RDY,@#CS1      ;RHCS1 SHOULD HAVE DEVICE AVAILABLE
8145                                                                  ;AND BE READY
8146  040772  001004                          BNE     1$              ;BRANCH IF GOOD
8147  040774  010137  001122                  MOV     R1,@#SBDADR     ;FAILING REGISTER
8148  041000  104026                          ERROR   26              ;RHCS1 IS IN ERROR
8149                                                                  ;DOES NOT HAVE DVA, RDY
8150  041002  000427                          BR      4$              ;BRANCH OUT
8151  041004  032737  004000  001702  1$:     BIT     #DVA,@#CS1      ;RHCS1 SHOULD HAVE DEVICE AVAILABLE
8152                                                                  ;AND BE READY
8153  041012  001004                          BNE     2$              ;BRANCH IF GOOD
8154  041014  010137  001122                  MOV     R1,@#SBDADR     ;FAILING REGISTER
8155  041020  104026                          ERROR   26              ;RHCS1 IS IN ERROR
8156                                                                  ;DOES NOT HAVE DVA, RDY
8157  041022  000417                          BR      4$              ;BRANCH OUT
8158  041024  032737  000200  001724  2$:     BIT     #DRY,@#DS1      ;RHDS1 SHOULD HAVE DPR,DRY
8159  041032  001004                          BNE     3$              ;BRANCH IF THERE
8160  041034  010337  001122                  MOV     R3,@#SBDADR     ;FAILING REGISTER RHDS1
8161  041040  104026                          ERROR   26              ;RHDS1 DOES NOT HAVE DPR,DRY
8162  041042  000407                          BR      4$              ;BRANCH OUT
8163  041044  032737  000400  001724  3$:     BIT     #DPR,@#DS1      ;RHDS1 SHOULD HAVE DPR,DRY
8164  041052  001004                          BNE     5$              ;BRANCH OUT AND CONTINUE IF THERE
8165  041054  010337  001122                  MOV     R3,@#SBDADR     ;FAILING REGISTER RHDS1
8166  041060  104026                          ERROR   26              ;RHDS1 DOES NOT HAVE DPR,DRY
8167
8168  041062  000207                  4$:     RTS     PC              ;RETURN TO TEST AND HALT
8169
8170  041064  062716  000006          5$:     ADD     #6,(SP)         ;ADJUST STACK TO GET OVER HALT IN TEST
8171  041070  000207                          RTS     PC              ;RETURN TO TEST AND CONTINUE TESTING
```

```
                                        ;       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)

041072  177777                  TIMCNT: 177777                  ;WAITING COUNT
041074  010046                  WAIT.T: MOV     R0,-(SP)                ;SAVE R0
041076  016600  000002                  MOV     2(SP),R0                ;GET ADRESS OF REG. ADDERSS
041102  010037  001176                  MOV     R0,@#$T_3               ;WAT PC+2 IN $TMP3
041106  162737  000002  001176          SUB     #2,@#$TMP3              ;WAT PC FOR TYPEOUT
041114  012037  001170                  MOV     (R0)+,@#$TMP0           ;WAIT REGISTER ADDRESS
041120  012037  001172                  MOV     (R0)+,@#$TMP1           ;WAIT ON BIT
041124  010066  000002                  MOV     R0,2(SP)                ;RESTORE RETURN ON STACK
041130  012600                          MOV     (SP)+,R0                ;RESTORE R0
041132  013737  041072  001174          MOV     @#TIMCNT,@#$TMP2        ;TEMPORARY COUNT
041140  033777  001172  140022  1$:     BIT     @#$TMP1,@$TMP0  ;IS REQUIRED BIT THERE?
041146  001021                          BNE     2$              ;BRANCH IF YES
041150  005337  001174                  DEC     @#$TMP2         ;COUNT
041154  001371                          BNE     1$              ;BRANCH IF NOT TIME UP
041156  013737  041072  001174          MOV     @#TIMCNT,@#$TMP2        ;TEMPORARY COUNT
041164  033777  001172  137776  3$:     BIT     @#$TMP1,@$TMP0  ;IS REQUIRED BIT THERE?
041172  001007                          BNE     2$              ;BRANCH IF YES
041174  005337  001174                  DEC     @#$TMP2         ;COUNT
041200  001371                          BNE     3$              ;BRANCH IF NOT TIME UP
041202  017737  137762  001126          MOV     @$TMP0,@#$BODAT ;REGISTER CONTENTS
041210  104016                          ERROR   16              ;WAITED ON BIT FAILED TO SET
041212  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

                                                .SBTTL  SAVE ROUTINE

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

041214                          SAVER:
041214  010146                          MOV     R1,-(SP)        ;;PUSH R1 ON STACK
041216  010246                          MOV     R2,-(SP)        ;;PUSH R2 ON STACK
041220  010346                          MOV     R3,-(SP)        ;;PUSH R3 ON STACK
```

```
8226  041222  012001                          MOV     (R0)+,R1        ;FROM
8227  041224  012002                          MOV     (R0)+,R2        ;TO
8228  041226  012003                          MOV     (R0)+,R3        ;NUMBER
8229  041230  013122                  1$:     MOV     @(R1)+,(R2)+    ;SAVE REGISTER CONTENTS
8230  041232  005303                          DEC     R3              ;COUNT
8231  041234  001375                          BNE     1$              ;BRANCH IF NOT DONE
8232  041236  012603                          MOV     (SP)+,R3        ;;POP STACK INTO R3
8233  041240  012602                          MOV     (SP)+,R2        ;;POP STACK INTO R2
8234  041242  012601                          MOV     (SP)+,R1        ;;POP STACK INTO R1
8235  041244  000200                          RTS     R0
8236
8237
8238
8239
8240                                          .SBTTL  WRITE CHECK ROUTINE
8241
8242                                          ;THIS IS A SUBROUTINE TO DO WRITE CHECK HEADER AND DATA
8243                                          ;CYLINDER 0, TRACK 1, SECTOR 1, KEYS 0
8244
8245                                          ;THESE ARE TO SET UP FOR DISKLESS USE ONLY
8246  041246  012737  010000  045152  WRCHHD: MOV     #FMT22,@#CYL    ;CYLINDER 0 FORMAT 16 BIT WORDS
8247  041254  112737  000001  045155          MOVB    #1,@#SECOTR+1   ;TRACK=1
8248  041262  112737  000001  045154          MOVB    #1,@#SECOTR     ;SECTOR=1
8249  041270  005037  045156                  CLR     @#KEY1          ;KEY1=0
8250  041274  005037  045160                  CLR     @#KEY2          ;KEY2=0
8251  041300  012767  000044  003724          MOV     #36.,DAWORD     ;NO OF DATA WORDS
8252  041306  005037  045162                  CLR     @#X             ;THIS IS A READ OPERATION
8253  041312  004537  041730                  JSR     R5,@#CRC        ;GO TO CALCULATE CRC
8254  041316  045152                          CYL
8255  041320  047052                          WCRC
8256
8257                                          ;THESE ARE REGULAR SETUPS
8258
8259  041322  004737  040522                  JSR     PC,@#CLDISK     ;SET UP GENERAL REGISTERS
8260                                                                  ;AND CLEAR DISK REGISTERS
8261  041326  012777  177730  140270          MOV     #-40.,@RHWC     ;36 DATA WORDS 4 HEADER WORDS
8262  041334  012777  003134  140264          MOV     #REINTO,@RHBA   ;STARTING ADDRESS OF READ BUFFER
8263  041342  112746  000001                  MOVB    #1,-(SP)        ;SECTOR=1
8264  041346  112766  000001  000001          MOVB    #1,1(SP)        ;TRACK=1 IN UPPER BYTE
8265  041354  012677  140256                  MOV     (SP)+,@RHDST    ;TRACK=1, SECTOR=1 IN RHDST
8266  041360  012777  014000  140254          MOV     #FMT22!ECI,@RHOF ;16 BIT WORDS
8267                                                                  ;ECC CORRECTION INHIBIT BECAUSE
8268                                                                  ;ECC LOGIC IS NOT CHECKED YET
8269  041366  005077  140252                  CLR     @RHCA           ;CYLINDER=0
8270  041372  004737  040556                  JSR     PC,@#CHECKT     ;CHECK THAT DVA,RDY,DPR,DRY = 1
8271  041376  104400  057176                  TYPE    ,CPHALT         ;AND THAT NO OTHERS = 1.  CANNOT CON-
8272                                                                  ;TINUE TESTING IF BOTH AREN'T TRUE
8273  041402  000000                          HALT                    ;STOP THE TEST
8274  041404  013711  002042                  MOV     @#WRCHDT,@R1    ;WRITE CHECK HEADER AND DATA=52
8275                                                                  ;INTO RHCS1
8276  041410  004737  045026                  JSR     PC,@#COMHD      ;WRITE CHECK HEADER AND DATA
8277                                                                  ;SAME AS READ HEADER AND DATA
8278
8279  041414  000207                          RTS     PC              ;RETURN TO WRITE CHECK TEST
```

```
                                        .SBTTL  COMPARE ROUTINE


                                ;THIS IS A SUBROUTINE TO COMPARE TWO BLOCKS IN MEMORY
                                ;R1 HAS GOOD DATA BUFFER ADDRESS
                                ;R2 HAS TEST DATA BUFFER ADDRESS
                                ;$TMP0 HAS ADDRESS OF RETURN ON ERROR TO PRINT HEADER
                                ;$TMP1 HAS ADDRESS OF RETURN ON ERROR TO PRINT DATA
                                ;R3 HAS NUMBER OF WORDS TO BE COMPARED
                                ;R4 HAS ONE MORE THAN NUMBER OF WORDS TO BE COMPARED

041416                          COMPAR:
041416  010146                          MOV     R1,-(SP)        ;;PUSH R1 ON STACK
041420  010246                          MOV     R2,-(SP)        ;;PUSH R2 ON STACK
041422  010346                          MOV     R3,-(SP)        ;;PUSH R3 ON STACK
041424  010446                          MOV     R4,-(SP)        ;;PUSH R4 ON STACK
041426  010546                          MOV     R5,-(SP)        ;;PUSH R5 ON STACK
041430  012001                          MOV     (R0)+,R1        ;ADDRESS OF GOOD DATA BUFFER
041432  012002                          MOV     (R0)+,R2        ;ADDRESS OF TEST DATA BUFFER
041434  012003                          MOV     (R0)+,R3        ;NO OF WORDS TO BE COMPARED
041436  012067  137526                  MOV     (R0)+,$TMP0     ;RETURN ON ERROR TO PRINT HEADER
041442  012067  137524                  MOV     (R0)+,$TMP1     ;RETURN ON ERROR TO PRINT DATA
041446  011000                          MOV     (R0),R0         ;RETURN ON NO ERROR
041450  010304                          MOV     R3,R4           ;NO OF WORDS TO BE COMPARED
041452  005204                          INC     R4
041454  010437  045272          1$:     MOV     R4,@#ERWORD     ;FOR ERROR WORD NO
041460  022122                          CMP     (R1)+,(R2)+     ;COMPARE GOOD WITH TEST DATA
041462  001426                          BEQ     3$              ;BRANCH IF GOOD

041464  014137  001124                  MOV     -(R1),@#SGDDAT  ;GOOD DATA
041470  014237  001126                  MOV     -(R2),@#SBDDAT  ;BAD DATA
041474  160337  045272                  SUB     R3,@#ERWORD     ;ERROR WORD NO.
041500  005737  001774                  TST     @#ERFLG$        ;ANY ERRORS ALREAY THERE
041504  001003                          BNE     2$              ;BRANCH IF YES
041506  004777  137456                  JSR     PC,@$TMP0       ;RETURN TO PRINT HEADER
041512  000402                          BR      5$              ;BRANCH TO AVOID PRINTING NEXT ERROR
041514  004777  137452          2$:     JSR     PC,@$TMP1       ;RETURN TO PRINT DATA
041520  022122                  5$:     CMP     (R1)+,(R2)+     ;UNDO -(R1) AND -(R2) FOR ERRORS
041522  013746  177570                  MOV     @#SWR,-(SP)     ;GET SWITCH SETTING
041526  042716  177177                  BIC     #^C600,(SP)     ;KEEP ONLY SWITCH 7 AND 8
041532  022726  000200                  CMP     #SW07,(SP)+     ;IS 7 SET AND 8 RESET
041536  001402                          BEQ     4$              ;BRANCH OUT IF YES
041540  005303                  3$:     DEC     R3              ;COUNT
041542  001344                          BNE     1$              ;BRANCH IF ALL NOT DEVICE
041544                          4$:
041544  012605                          MOV     (SP)+,R5        ;;POP STACK INTO R5
041546  012604                          MOV     (SP)+,R4        ;;POP STACK INTO R4
041550  012603                          MOV     (SP)+,R3        ;;POP STACK INTO R3
041552  012602                          MOV     (SP)+,R2        ;;POP STACK INTO R2
041554  012601                          MOV     (SP)+,R1        ;;POP STACK INTO R1
```

```
8334  041556  000200                          RTS     R0              ;RETURN TO MAIN PROGRAM
8335
8336
8337
8338
8339                                          ;THIS IS A SUBROUTINE TO DO WRITE CHECK DATA
8340                                          ;CYLINDER 0, TRACK 1, SECTOR 1, KEYS 0
8341
8342                                          ;THESE ARE TO SET UP FOR DISKLESS USE ONLY
8343
8344  041560  012737  010000  045152  WRCHDA: MOV     #FMT22,@#CYL    ;CYLINDER 0 FORMAT 16 BIT WORDS
8345  041566  112737  000001  045155          MOVB    #1,@#SECOTR+1   ;TRACK=1
8346  041574  112737  000001  045154          MOVB    #1,@#SECOTR     ;SECTOR=1
8347  041602  005037  045156                  CLR     @#KEY1          ;KEY1=0
8348  041606  005037  045160                  CLR     @#KEY2          ;KEY2=0
8349  041612  012737  000040  045232          MOV     #32.,@#DAWORD   ;NO OF DATA WORDS
8350  041620  005037  045162                  CLR     @#X             ;THIS IS A READ OPERATION
8351
8352  041624  004537  041730                  JSR     R5,@#CRC        ;GO TO CALCULATE CRC
8353  041630  045152                          CYL
8354  041632  047052                          WCRC
8355
8356                                          ;THESE ARE REGULAR SETUPS
8357
8358  041634  004737  040522                  JSR     PC,@#CLDISK     ;SET UP GENERAL REGISTERS
8359                                                                  ;AND CLEAR DISK REGISTERS
8360
8361  041640  012777  177740  137756          MOV     #-32.,@RHWC     ;36 DATA WORDS 4 HEADER WORDS
8362  041646  012777  003134  137752          MOV     #REINTO,@RHBA   ;STARTING ADDRESS OF READ BUFFER
8363  041654  112746  000001                  MOVB    #1,-(SP)        ;SECTOR=1
8364  041660  112766  000001  000001          MOVB    #1,1(SP)        ;TRACK=1 IN UPPER BYTE
8365  041666  012677  137744                  MOV     (SP)+,@RHDST    ;TRACK=1, SECTOR=1 IN RHDST
8366  041672  012777  014000  137742          MOV     #FMT22!ECI,@RHOF        ;16 BIT WORDS
8367                                          ;ECC CORRECTION INHIBIT BECAUSE
8368                                          ;ECC LOGIC IS NOT CHECKED YET
8369  041700  005077  137740                  CLR     @RHCA           ;CYLINDER=0
8370  041704  004737  040556                  JSR     PC,@#CHECKT     ;CHECK THAT DVA,RDY,DPR,DRY = 1
8371  041710  104400  057176                  TYPE    ,CPHALT         ;AND THAT NO OTHERS = 1.  CANNOT CON-
8372                                          ;TINUE TESTING IF BOTH AREN'T TRUE
8373  041714  000000                          HALT                    ;STOP THE TEST
8374  041716  013711  002040                  MOV     @#WRCHEK,@R1    ;WRITE CHECK DATA=50 INTO RHCS1
8375  041722  004737  045026                  JSR     PC,@#COMHD      ;WRITE CHECK HEADER AND DATA
8376                                                                  ;SAME AS READ HEADER AND DATA
8377
8378  041726  000207                          RTS     PC              ;RETURN TO WRITE CHECK TEST
```

```
                                        .SBTTL  CRC GENERATION ROUTINE


                                        ;THIS IS A SUBROUTINE TO CALCULATE CRC FOR THE FOUR
                                        ;HEADER WORDS AND STORE THEM IN "WCRC" AND "GCRC"
                                        ;R1 - REGISTER FOR CRC, INCREMENTED CRC VALUE IS HERE
                                        ;R2 - THIS HAS BIT POSITION 2 VALUE C
                                        ;R3 - THIS HAS BIT POSITION 16 I.E. OUTPUT BIT VALUE B
                                        ;R4 - THIS HAS BIT POSITION 15 VALUE E
                                        ;$TMP0 - NUMBER OF WORDS
                                        ;$TMP2 - NUMBER OF BITS PER WORD = 16
                                        ;$TMP3 - TEMPORARY REG.
                                        ;$TMP4 - TEMPORARY REG TO TRANSFER CARRY
                                        ;$TMP5 - THIS HAS DATA BIT VALUE D

                                        ;FETCH DATA BIT D
                                        ;B = D XOR 16
                                        ;C = B XOR 2
                                        ;E = B XOR 15
                                        ;ROTATE RIGHT ONE POSITION
                                        ;B GOES TO POSITION 1
                                        ;C GOES TO POSITION 3
                                        ;E GOES TO POSITION 16
                                        ;REPET 64 TIMES
                                        ;CALL   JSR     R5,@#CRC
                                        ;X      ;FIRST LOCATION AT
                                        ;Y      ;PUT CRC IN WCRC FOR READ GCRC FOR WRITE

041730                          CRC:
041730  010046                  MOV     R0,-(SP)        ;;PUSH R0 ON STACK
041732  012500                  MOV     (R5)+,R0        ;GET POINTER TO CYL NO.
041734  010146                  MOV     R1,-(SP)        ;;PUSH R1 ON STACK
041736  010246                  MOV     R2,-(SP)        ;;PUSH R2 ON STACK
041740  010346                  MOV     R3,-(SP)        ;;PUSH R3 ON STACK
041742  010446                  MOV     R4,-(SP)        ;;PUSH R4 ON STACK
041744  005001                  CLR     R1              ;CLEAR WORKING LOCATION
041746  005037  001202          CLR     @#$TMP5
041752  012737  000004  001170  MOV     #4,@#$TMP0      ;WORD COUNT
041760  012037  001176  16$:    MOV     (R0)+,@#$TMP3   ;TEMPORARY WORD STORAGE
041764  012767  000020  137202  MOV     #16.,$TMP2      ;BIT COUNT
041772  013737  001176  001200  MOV     @#$TMP3,@#$TMP4 ;TEMPORARY WORD STORAGE
042000  006037  001176  15$:    ROR     @#$TMP3         ;GET LSB INTO "C"
042004  006037  001202          ROR     @#$TMP5         ;GET ABOVE "C" INTO $TMP5
042010  032701  000001          BIT     #BIT0,R1        ;IS POSITION 15 HIGH
042014  001403                  BEQ     1$              ;BRANCH IF POSITION 16 LOW
042016  012703  100000          MOV     #BIT15,R3       ;GET POSITION 16
042022  000401                  BR      2$
042024  005003          1$:     CLR     R3              ;GET POSITION 16
042026  063703  001-32  2$:     ADD     @#$TMP5,R3      ;XOR POSITION 16 WITH D
                                                        ;TO GIVE B
042032  032701  040000          BIT     #BIT14,R1       ;IS POSITION 2 HIGH
042036  001403                  BEQ     3$              ;BRANCH IF POSITION 2 LOW
042040  012702  100000          MOV     #BIT15,R2       ;GET POSITION 2
```

```
8433  042044  000401                        BR      4$
8434  042046  005002              3$:       CLR     R2              ;GET POSITION 2
8435  042050  060302              4$:       ADD     R3,R2           ;XOR B WITH POSITION 2
8436                                                                ;TO GIVE C
8437  042052  032701  000002              BIT     #BIT1,R1        ;IS POSITION 15 HIGH
8438  042056  001403                        BEQ     5$              ;BRANCH IF POSITION 15 LOW
8439  042060  012704  100000              MOV     #BIT15,R4       ;GET POSITION 15
8440  042064  000401                        BR      6$
8441  042066  005004              5$:       CLR     R4              ;GET POSITION 15
8442  042070  060304              6$:       ADD     R3,R4           ;XOR POSITION 15 WITH B
8443                                                                ;TO GIVE E
8444  042072  006037  001200              ROR     @#STMP4         ;GET LSB INTO "C"
8445  042076  006001                        ROR     R1              ;GET ABOVE C INTO R1
8446  042100  005703                        TST     R3              ;TEST B
8447  042102  100403                        BMI     7$              ;BRANCH IF B=1
8448  042104  042701  100000              BIC     #BIT15,R1       ;SET B IN POSITION 1
8449  042110  000402                        BR      10$
8450  042112  052701  100000              BIS     #BIT15,R1       ;SET B IN POSITION 1
8451  042116  005702              10$:      TST     R2              ;TEST C
8452  042120  100403                        BMI     11$             ;BRANCH IF C=1
8453  042122  042701  020000              BIC     #BIT13,R1       ;GET C IN POSITION 3
8454  042126  000402                        BR      12$
8455  042130  052701  020000              BIS     #BIT13,R1       ;GET C IN POSITION 3
8456  042134  005704              12$:      TST     R4              ;TEST E
8457  042136  100403                        BMI     13$             ;BRANCH IF E=1
8458  042140  042701  000001              BIC     #BIT0,R1        ;GET E IN POSITION 16
8459  042144  000402                        BR      14$
8460  042146  052701  000001              BIS     #BIT0,R1        ;GET E IN POSITION 16
8461  042152  005337  001174              DEC     @#STMP2         ;BIT COUNTER
8462  042156  001310                        BNE     15$             ;BRANCH IF 16 NOT DONE
8463  042160  005337  001170              DEC     @#STMP0         ;WORD COUNTER
8464  042164  001275                        BNE     16$             ;BRANCH IF 4 NOT DONE
8465  042166  010135                        MOV     R1,@(R5)+       ;PUT CRC WHERE DESIRED
8466  042170  012604                        MOV     (SP)+,R4        ;;POP STACK INTO R4
8467  042172  012603                        MOV     (SP)+,R3        ;;POP STACK INTO R3
8468  042174  012602                        MOV     (SP)+,R2        ;;POP STACK INTO R2
8469  042176  012601                        MOV     (SP)+,R1        ;;POP STACK INTO R1
8470  042200  012600                        MOV     (SP)+,R0        ;;POP STACK INTO R0
8471  042202  000205                        RTS     R5
8472
8473
8474
8475
8476                                                                ;THIS IS A SUBROUTINE TO SET UP THE SIMULATOR DISK FOR
8477                                                                ;CYLINDER 0 (16 BITS PER WORD)
8478                                                                ;TRACK 1, SECTOR 1
8479                                                                ;KEY1 1
8480                                                                ;KEY2 1
8481                                                                ;CRC THROUGH THE JSR R5,@#CRC
8482                                                                ;256 WORDS OF 177400
8483
8484                                                                ;CALL   JSR     PC,@#SETDSK
8485
8486  042204                    SETDSK:
```

```
8487  042204  010046                          MOV     R0,-(SP)        ;;PUSH R0 ON STACK
8488  042206  010146                          MOV     R1,-(SP)        ;;PUSH R1 ON STACK
8489  042210  010246                          MOV     R2,-(SP)        ;;PUSH R2 ON STACK
8490  042212  012700  177400                  MOV     #177400,R0      ;DATA IN THE DISK
8491  042216  012701  000400                  MOV     #256.,R1        ;COUNTER
8492  042222  012702  047070                  MOV     #DISK,R2        ;START OF SIMULATOR DISK
8493  042226  010022                  1$:     MOV     R0,(R2)+        ;MOVE IN DATA
8494  042230  005301                          DEC     R1              ;COUNT FOR 256
8495  042232  001375                          BNE     1$              ;BRANCH IF 256 NOT COMPLETE
8496  042234  012701  000021                  MOV     #17.,R1         ;2 ECC WORDS, 1 DATA GAP
8497                                                                  ;14 TOLERANCE GAP
8498  042240  005022                  2$:     CLR     (R2)+           ;CLEAR ECC,DATA GAP AND
8499                                                                  ;TOLERANCE GAP
8500  042242  005301                          DEC     R1              ;COUNT
8501  042244  001375                          BNE     2$              ;BRANCH IF NOT COMPLETE
8502
8503                                          ;NOW SET UP FOR DISKLESS USE
8504
8505  042246  012737  010000  045152          MOV     #FMT22,@#CYL    ;CYLINDER 0 (16 BIT WORDS)
8506  042254  112737  000001  045155          MOVB    #1,@#SECOTR+1   ;TRACK=1
8507  042262  112737  000001  045154          MOVB    #1,@#SECOTR     ;SECTOR=1
8508  042270  012737  000001  045156          MOV     #1,@#KEY1       ;KEY1=1
8509  042276  012737  000001  045160          MOV     #1,@#KEY2       ;KEY2=1
8510  042304  016737  136070  045232          MOV     256.,@#DATAWORD  ;NO. OF DATA WORDS
8511  042312  004537  041730                  JSR     R5,@#CRC        ;GO TO CALCULATE CRC
8512  042316  045152                          CYL                     ;FIRST CRC WORD
8513  042320  047052                          WCRC                    ;PUT CALCULATED CRC
8514  042322  012602                          MOV     (SP)+,R2        ;;POP STACK INTO R2
8515  042324  012601                          MOV     (SP)+,R1        ;;POP STACK INTO R1
8516  042326  012600                          MOV     (SP)+,R0        ;;POP STACK INTO R0
8517  042330  000207                          RTS     PC
```

```
                                        ;THIS IS A SUBROUTINE TO CHECK HEADER COMPARE ERROR
                                        ;(BIT #7) AND CRC ERROR (BIT #8)

                                        ;CALL JSR R0,@#HCCRCE

                                        ;       COM             ;COMMAND-READ HEADER AND DATA
                                        ;                       ;       -WRITE DATA
                                        ;       C               ;CYLINDER
                                        ;       S               ;SECTOR
                                        ;       T               ;TRACK
                                        ;       -N.             ;WORD COUNT
                                        ;       B               ;RHBA BUFFER START
                                        ;       X               ;1=WRITE DATA 0=READ
                                        ;       H               ;H=1 HEADER CHECK, H=0 CRC CHECK

042332  010037  002002          HCCRCE: MOV     R0,@#PCJSR      ;SAVE PC OF JSR+4
042336  162737  000004  002002          SUB     #4,@#PCJSR      ;GET PC OF JSR
042344  004737  040522                  JSR     PC,@#CLDISK     ;INIT AND SETUP GENERAL REG.
042350  004737  040556                  JSR     PC,@#CHECK1     ;CHECK THAT DVA,RDY,DPR,DRY = 1
042354  104400  057176                  TYPE    ,CPHALT         ;AND THAT NO OTHERS = 1.  CANNOT CON-
                                                                ;TINUE TESTING IF BOTH AREN'T TRUE
042360  000000                          HALT                    ;STOP THE TEST
042362  011037  001202                  MOV     (R0),@#STMPS    ;SAVE COMMAND
042366  012011                          MOV     (R0)+,@R1       ;COMMAND
042370  012077  137250                  MOV     (R0)+,@RHCA     ;CYLINDER
042374  112046                          MOVB    (R0)+,-(SP)     ;SECTOR
042376  105720                          TSTB    (R0)+           ;UP DATE R0
042400  112066  000001                  MOVB    (R0)+,1(SP)     ;TRACK
042404  105720                          TSTB    (R0)+           ;UPDATE R0
042406  012677  137224                  MOV     (SP)+,@RHDST    ;TRACK SECTOR
042412  012077  137206                  MOV     (R0)+,@RHWC     ;NO. OF DATA WORDS +4 HEADER
                                                                ;IF A READ HEADER AND DATA
042416  012077  137204                  MOV     (R0)+,@RHBA     ;STARTING ADDRESS OF BUFFER
042422  012037  045162                  MOV     (R0)+,@#X       ;X=0 READ HEADER AND DATA
                                                                ;X=1 WRITE DATA
042426  012777  014000  137206          MOV     #FMT22!ECI,@RHOF ;16 BITS PER WORD
                                                                ;ECC CORRECTION INHIBIT
042434  005037  001774                  CLR     @#ERFLG$        ;CLEAR ERROR FLAG
042440  004737  045026                  JSR     PC,@#COMHD      ;COMMAND

                                        ;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.

042444  004737  040126                  JSR     PC,@#PUTREG     ;SAVE REGISTERS
042450  005737  001774                  TST     @#ERFLG$        ;ANY ERRORS ALREADY THERE
042454  001034                          BNE     10$             ;BRANCH IF YES
042456  005737  045162                  TST     @#X             ;IS THIS A READ
042462  001015                          BNE     3$              ;IF A WRITE DATA BRANCH
```

```
8571
8572                                        ;NOW THE READ BUFFER WILL BE CHECKED
8573                                        ;HEADER SHOULD BE COMPLETELY READ AS WRITTEN
8574                                        ;NO DATA WORDS SHOULD BE READ
8575                                        ;REINTO BUFFER HAS BEEN FILLED WITH 0
8576                                        ;WRFROM BUFFER HAS BEEN FILLED WITH EXPECTED DATA
8577
8578  042464  004037  041416                JSR     R0,@#COMPAR     ;CHECK
8579  042470  002070                        WRFROM                  ;GOOD DATA
8580  042472  003134                        REINTO                  ;TEST BUFFER
8581  042474  000400                        256.                    ;4 HEADER 252 DATA
8582  042476  042504                        1$                      ;RETURN POINT FOR ERROR HEADER
8583  042500  042510                        2$                      ;RETURN POINT FOR ERROR DATA
8584  042502  042546                        10$                     ;RETURN FOR GOOD COMPARISON
8585  042504  104004              1$:       ERROR   4               ;READ NEXT ERROR 5
8586  042506  000207                        RTS     PC              ;RETURN TO COMPARISON SUBROUTINE
8587  042510  104005              2$:       ERROR   5               ;WORD NO 1 THRU 4 ARE
8588                                                                ;HEADER WORDS AND HENCE
8589                                                                ;SHOULD BE READ AS WRITTEN ON
8590                                                                ;DISK, WORD NOS. 5 ONWARDS
8591                                                                ;SHOULD NOT BE READ AND HENCE
8592                                                                ;READ INTO BUFFER
8593                                                                ;SHOULD BE UNCHANGED
8594  042512  000207                        RTS     PC              ;RETURN TO COMPARISON
8595
8596  042514  000414                        BR      10$             ;JUMP OUT
8597
8598                                        ;NOW THE DISK WILL BE CHECKED
8599                                        ;NO DATA SHOULD BE WRITTEN
8600                                        ;REINTO BUFFER HAS BEEN FILLED WITH EXPECTED DATA
8601                                        ;DISK HAS BEEN FILLED WITH 177400
8602                                        ;WRFROM HAS BEEN FILLED WITH 125252
8603
8604  042516  004037  041416        3$:       JSR     R0,@#COMPAR     ;CHECK
8605  042522  003134                        REINTO                  ;GOOD DATA BUFFER
8606  042524  047070                        DISK                    ;TEST BUFFER
8607  042526  000400                        256.
8608  042530  042536                        4$                      ;RETURN POINT FOR ERROR HEADER
8609  042532  042542                        5$                      ;RETURN POINT FOR ERROR DATA
8610  042534  042546                        10$                     ;RETURN POINT FOR GOOD COMPARISON
8611  042536  104004              4$:       ERROR   4               ;READ NEXT ERROR 5
8612  042540  000207                        RTS     PC              ;RETURN TO COMPARISON SUBROUTINE
8613  042542  104005              5$:       ERROR   5               ;WORD NO ARE ALL DATA
8614                                                                ;WORDS THE SHOULD NOT
8615                                                                ;HAVE BEEN CHANGED BY THE
8616                                                                ;WRITE COMMAND
8617  042544  000207                        RTS     PC              ;RETURN TO COMPARISON SUBROUTINE
8618  042546  005720              10$:      TST     (R0)+           ;IS THIS A HCRC ON HCE CHECK?
8619  042550  001442                        BEQ     6$              ;BRANCH IF HCRC
8620  042552  022737  000072  001202        CMP     #72,@#STMPS     ;IS THIS A READ COMMAND
8621  042560  001417                        BEQ     11$             ;BRANCH IF YES
8622  042562  017737  137046  001126        MOV     @RHER1,@#SBDDAT ;TEST DATA
8623  042570  022737  000200  001126        CMP     #HCE,@#SBDDAT   ;ONLY HEADER COMPARE BIT?
8624                                                                ;SHOULD BE SET
```

```
 8625  042576  001470                          BEQ     7$                ;BRANCH IF GOOD
 8626  042600  013737  001634  040172          MOV     @#RHER1,@#REGADR  ;REGISTER ADDRESS RHER1
 8627  042606  012737  000200  001124          MOV     #HCE,@#$GDDAT     ;GOOD DATA
 8628  042614  104027                          ERROR   27                ;AFTER AN ERROR ON THE
 8629                                                                    ;HEADER ONLY HCE SHOULD
 8630  042616  000460                          BR      7$                ;BE SET
 8631  042620                          11$:
 8632  042620  017737  137010  001126          MOV     @RHER1,@#$BDDAT   ;TEST DATA
 8633  042626  022737  100200  001126          CMP     #DCK!HCE,@#$BDDAT ;ONLY HEADER COMPARE BIT?
 8634                                                                    ;SHOULD BE SET
 8635                                                                    ;DCK IS SET BECAUSE ECC IS NOT READ
 8636  042634  001451                          BEQ     7$                ;BRANCH IF GOOD
 8637  042636  013737  001634  040172          MOV     @#RHER1,@#REGADR  ;REGISTER ADDRESS RHER1
 8638  042644  012737  100200  001124          MOV     #DCK!HCE,@#$GDDAT       ;GOOD DATA
 8639  042652  104027                          ERROR   27                ;AFTER AN ERROR ON THE
 8640                                                                    ;HEADER ONLY HCE SHOULD
 8641  042654  000441                          BR      7$                ;BE SET
 8642  042656  022737  000072  001202  6$:     CMP     #72,@#STMPS       ;IS THIS A READ COMMAND?
 8643  042664  001417                          BEQ     12$               ;BRANCH IF A READ
 8644  042666  017737  136742  001126          MOV     @RHER1,@#$BDDAT   ;TEST DATA
 8645  042674  022737  000400  001126          CMP     #HCRC,@#$BDDAT    ;ONLY CRC ERROR SHOULD BE THERE
 8646  042702  001426                          BEQ     7$
 8647  042704  013737  001634  040172          MOV     @#RHER1,@#REGADR  ;REG. ADDR = RHER1
 8648  042712  012737  000400  001124          MOV     #HCRC,@#$GDDAT    ;GOOD DATA
 8649  042720  104027                          ERROR   27                ;AFTER A CRC ERROR ONLY CRC
 8650                                                                    ;SHOULD BE SET
 8651  042722  000416                          BR      7$                ;BRANCH OUT
 8652  042724  017737  136704  001126  12$:    MOV     @RHER1,@#$BDDAT   ;TEST DATA
 8653
 8654  042732  022737  100400  001126          CMP     #DCK!HCRC,@#$BDDAT;HCRC AND DCK SHOULD BE SET
 8655                                                                    ;DCK IS SET BECAUSE ECC IS NOT READ
 8656  042740  001407                          BEQ     7$                ;BRANCH IF GOOD
 8657  042742  012737  100400  001124          MOV     #DCK!HCRC,@#$GDDAT;GOOD DATA
 8658  042750  013737  001634  040172          MOV     @#RHER1,@#REGADR;FAILING REGISTER RHER1
 8659  042756  104027                          ERROR   27                ;AFTER A CRC ERROR ON A READ
 8660                                                                    ;DCK AND HCRC SHOULD BE SET
 8661                                                                    ;DCK IS SET BECAUSE ECC IS NOT READ
 8662  042760  000200                  7$:     RTS     R0                ;RETURN TO MAIN TEST
 8663
 8664
 8665
 8666                                          ;THIS IS A SUBROUTINE TO LEAVE AT THE MIDDLE OF
 8667                                          ;A WRITE HEADER AND DATA COMMAND
 8668                                          ;IT TRYS TO GET SECTOR 10, TRACK 0, CYLINDER 0
 8669                                          ;BUT COMES OUT AFTER ONE SECTOR
 8670                                          ;THE COMMAND OS JSR PC,@#MIDDLE
 8671                                          ;BAI IS SET
 8672
 8673  042762                          MIDDLE:
 8674  042762  010046                          MOV     R0,-(SP)          ;;PUSH R0 ON STACK
 8675  042764  010146                          MOV     R1,-(SP)          ;;PUSH R1 ON STACK
 8676  042766  013777  002046  136636          MOV     @#WRIFOR,@RHCS1   ;WRITE HEADER AND DATA=62
 8677                                                                    ;IN RHCS1
 8678  042774  012777  177766  136622          MOV     #-10.,@RHWC       ;10 WORDS
```

```
8679  043002  012777  002070  136616          MOV     #WRFROM,@RHBA   ;BUS ADDRESS=WRFROM
8680  043010  012777  000010  136620          MOV     #10,@RHDST      ;DESIRED TRACK=0 SECTOR=10
8681  043016  052777  000010  136604          BIS     #BAI,@RHCS2     ;BUS ADDRESS INCREMENT INHIBIT
8682  043024  012777  010000  136610          MOV     #FMT22,@RHOF    ;FORMAT 16 BIT WORDS
8683  043032  005077  136606                  CLR     @RHCA           ;CYLINDER=0
8684  043036  012737  000001  043064          MOV     #1,@#MID        ;SECTOR IS SET TO 1 SO THAT
8685                                                                  ;WE CAN GET OUT AT THE
8686                                                                  ;MIDDLE OF AN OPERATION
8687                                                                  ;LOOKING FOR SECTOR 10
8688  043044  012777  000001  136600          MOV     #DMD,@RHMR      ;SET DIAGNOSTIC MODE
8689  043052  052777  000001  136552          BIS     #GO,@RHCS1              ;GO TO RHCS1 WITH 62
8690  043060  004137  051204                  JSR     R1,@#SEARCH
8691  043064  000000                  MID:    .WORD   0               ;SECTOR
8692  043066  012601                          MOV     (SP)+,R1        ;;POP STACK INTO R1
8693  043070  012600                          MOV     (SP)+,R0        ;;POP STACK INTO R0
8694  043072  000207                          RTS     PC
8695
8696
8697
8698                                          .SBTTL  JAM CURRENT CYLINDER ROUTINE
8699
8700
8701                                  ;THIS SUBROUTINE WILL CHANGE THE CURRENT CYLINDER REGISTER
8702                                  ;THIS IS DONE BY GIVING A SEEK COMMAND THEN AN INIT
8703                                  ;WHICH WILL LOAD THE CURRENT CYLINDER WITH THE DESIRED CYLINDER VALUE
8704                                  ;
8705                                  ;CALL IS
8706                                  ;JSR    R0,@#MAKECYL
8707                                  ;XC             ;DESIRED VALUE OF CURRENT CYLINDER
8708
8709
8710  043074                  MAKECYL:
8711  043074  010546                          MOV     R5,-(SP)        ;;PUSH R5 ON STACK
8712  043076  010037  002002                  MOV     R0,@#PCJSR      ;PC OF JSR+4
8713  043102  162737  000004  002002          SUB     #4,@#PCJSR      ;SAVE PC OF JSR
8714  043110  012005                          MOV     (R0)+,R5        ;GETTING READY TO FILL DESIRED CYLINDER
8715  043112  010577  136526                  MOV     R5,@RHCA        ;FILL DESIRED CYLINDER REGISTER
8716  043116  005077  136514                  CLR     @RHDST          ;MAKE SURE DESIRED SECTOR TRACK IS NOT ILLEGAL
8717  043122  013777  002054  136502          MOV     @#SEECOM,@RHCS1 ;FILL SEEK COMMAND
8718  043130  012777  000001  136514          MOV     #DMD,@RHMR      ;SET DIAGNOSTIC MODE
8719  043136  052777  000001  136466          BIS     #GO,@RHCS1      ;GO TO SEEK
8720  043144  000240                          NOP                     ;ALLOW TIME FOR SEEK TO HANG UP
8721  043146  000240                          NOP                     ;ALLOW TIME FOR SEEK TO HANG UP
8722  043150  000240                          NOP                     ;ALLOW TIME FOR SEEK TO HANG UP
8723  043152  000240                          NOP                     ;ALLOW TIME FOR SEEK TO HANG UP
8724  043154  004737  040522                  JSR     PC,@#CLDISK     ;GIVE INIT
8725  043160  017737  136504  001126          MOV     @RHCC,@#$BDDAT  ;TEST DATA
8726  043166  020537  001126                  CMP     R5,@#$BDDAT     ;COMPARE CURRENT CYLINDER
8727  043172  001406                          BEQ     1$              ;BRANCH IF GOOD
8728  043174  010537  001124                  MOV     R5,@#$GDDAT     ;GOOD VALUE OF RHCC
8729  043200  013737  001670  040172          MOV     @#RHCC,@#REGADR ;FAILING REGISTER ADDRESS
8730  043206  104030                          ERROR   30              ;CURRENT CYLINDER DOES NOT MATCH DESIRED CYLINDER
8731                                                                  ;REGISTER AFTER A SEEK AND AN INIT
8732  043210                  1$:
```

```
8733  043210  012605                          MOV     (SP)+,R5        ;;POP STACK INTO R5
8734  043212  000200                          RTS     R0
8735
8736
8737                                          .SBTTL  ECC GENERATION AND COMPARISON ROUTINE
8738
8739
8740
8741
8742                                  ;THIS SUBROUTINE GENERATES AND TESTS ECC
8743                                  ;CALL    JSR PC,ECTEST
8744
8745
8746
8747
8748
8749          100000                  PIE1    =100000
8750          040000                  PIE2    =40000
8751          020000                  PIE3    =20000
8752          010000                  PIE4    =10000
8753          004000                  PIE5    =4000
8754          002000                  PIE6    =2000
8755          001000                  PIE7    =1000
8756          000400                  PIE8    =400
8757          000200                  PIE9    =200
8758          000100                  PIE10   =100
8759          000040                  PIE11   =40
8760          000020                  PIE12   =20
8761          000010                  PIE13   =10
8762          000004                  PIE14   =4
8763          000002                  PIE15   =2
8764          000001                  PIE16   =1
8765          100000                  PIE17   =100000
8766          040000                  PIE18   =40000
8767          020000                  PIE19   =20000
8768          010000                  PIE20   =10000
8769          004000                  PIE21   =4000
8770          002000                  PIE22   =2000
8771          001000                  PIE23   =1000
8772          000400                  PIE24   =400
8773          000200                  PIE25   =200
8774          000100                  PIE26   =100
8775          000040                  PIE27   =40
8776          000020                  PIE28   =20
8777          000010                  PIE29   =10
8778          000004                  PIE30   =4
8779          000002                  PIE31   =2
8780          000001                  PIE32   =1
8781
8782  043214  000000                  ECDATA: 0               ;DATA BIT FOR ECC
8783                                                          ;IF ALL ONES THEN CURRENT BIT IS A ONE
8784                                                          ;IF ZERO THEN CURRENT BIT IS A ZERO
8785
8786  043216  000000                  GECC1:  0               ;LOW ORDER ECC WORD TO BE GENERATED HERE
```

```
 8787                                                    ;=R1
 8788
 8789  043220  000000          GECC2:  0                 ;HIGH ORDER ECC WORD TO BE GENERATED HERE
 8790                                                    ;=R2
 8791
 8792  043222  000000          TSECCG: 0                 ;IF =177777 GENERATE AND TEST ECC FOR THIS BIT
 8793                                                    ;IF =0 DO NOT GENERATE AND TEST ECC FOR THIS BIT
 8794
 8795  043224  113713          NCODE:  38859.            ;N-CODE WORD
 8796  043226  000000          NCOUNT: 0                 ;TEMPORARY N CODE
 8797  043230  000000          POSITI: 0                 ;POSITION REGISTER
 8798  043232  010041          HARDER: 4129.             ;HARD ERROR COUNT
 8799                                                    ;TRUE COUNT IS 4128 BUT AS COMPARES ARE
 8800                                                    ;DONE ONE STAGE LATER SO 4129
 8801  043234  000000          DATENV: 0                 ;DATA ENVELOPE FOR TYPE OUT
 8802                                                    ;MAX FOR WRITE IS 4096
 8803                                                    ;MAX FOR READ IS 4128
 8804  043236  000000          ZCODE:  0                 ;LEADING ZEROS ENVELOPE FOR TYPE OUT
 8805                                                    ;THIS IS SHUT OFF WHEN POSITION COUNTER
 8806                                                    ;IN ENABLED
 8807                                                    ;MAX COUNT IS 38859
 8808
 8809
 8810
 8811  043240  000000          HADTMP: 0                 ;TEMPORARY HARD ERROR COUNT
 8812  043242  000000          P3:     0
 8813  043244  000000          P12:    0
 8814  043246  000000          P22:    0
 8815  043250  000000          P24:    0
 8816
 8817
 8818
 8819
 8820
 8821  043252                  ECTEST:
 8822  043252  010046                  MOV     R0,-(SP)          ;;PUSH R0 ON STACK
 8823  043254  010146                  MOV     R1,-(SP)          ;;PUSH R1 ON STACK
 8824  043256  010246                  MOV     R2,-(SP)          ;;PUSH R2 ON STACK
 8825  043260  010346                  MOV     R3,-(SP)          ;;PUSH R3 ON STACK
 8826  043262  010446                  MOV     R4,-(SP)          ;;PUSH R4 ON STACK
 8827  043264  010546                  MOV     R5,-(SP)          ;;PUSH R5 ON STACK
 8828  043266  013701  043216          MOV     @#GECC1,R1                ;ECC1 WORD
 8829  043272  013702  043220          MOV     @#GECC2,R2                ;ECC2 WORD
 8830  043276  005737  043214          TST     @#ECDATA                  ;IS CURRENT BIT A ONE
 8831  043302  001406                  BEQ     2$                        ;BRANCH IF CURRENT DATA D=0
 8832
 8833                                  ;IF CARRY IS NOT ZERO THEN D=1
 8834                                  ;INVERT X32 TO GIVE R0
 8835
 8836  043304  010103          1$:     MOV     R1,R3
 8837  043306  052703  177776          BIS     #^CPIE32,R3
 8838  043312  005103                  COM     R3
 8839  043314  010300                  MOV     R3,R0
 8840  043316  000404                  BR      3$
```

```
8841
8842                                            ;IF CARRY IS ZERO THEN D=0
8843                                            ;X32 BECOMES R0
8844  043320  010103                    2$:     MOV     R1,R3
8845  043322  042703  177776                    BIC     #^CPIE32,R3
8846  043326  010300                            MOV     R3,R0
8847
8848  043330  000241                    3$:     CLC
8849  043332  006000                            ROR     R0
8850  043334  006000                            ROR     R0
8851  043336  005700                            TST     R0
8852  043340  001462                            BEQ     10$             ;BRANCH IF R0=0
8853                                            ;INVERT X2
8854
8855  043342  010203                            MOV     R2,R3
8856  043344  052703  137777                    BIS     #^CPIE2,R3
8857  043350  005103                            COM     R3
8858  043352  010337  043242                    MOV     R3,@#P3
8859  043356  006237  043242                    ASR     @#P3
8860
8861                                            ;INVERT X11
8862
8863
8864  043362  010203                            MOV     R2,R3
8865  043364  052703  177737                    BIS     #^CPIE11,R3
8866  043370  005103                            COM     R3
8867  043372  010337  043244                    MOV     R3,@#P12
8868  043376  006237  043244                    ASR     @#P12
8869
8870                                            ;INVERT X21
8871
8872  043402  010103                            MOV     R1,R3
8873  043404  052703  173777                    BIS     #^CPIE21,R3
8874  043410  005103                            COM     R3
8875  043412  010337  043246                    MOV     R3,@#P22
8876  043416  006237  043246                    ASR     @#P22
8877
8878                                            ;INVERT X23
8879
8880  043422  010103                            MOV     R1,R3
8881  043424  052703  176777                    BIS     #^CPIE23,R3
8882  043430  005103                            COM     R3
8883  043432  010337  043250                    MOV     R3,@#P24
8884  043436  006237  043250                    ASR     @#P24
8885
8886                                            ;NOW THAT R0 FOR POSITION 1
8887                                            ;         F3 FOR POSITION 3
8888                                            ;        P12 FOR POSITION 12
8889                                            ;        P22 FOR POSITION 22
8890                                            ;        P24 FOR POSITION 24
8891                                            ;ARE KNOWN THE ROTATE WILL BE DONE AND
8892                                            ;THESE BITS JAMED IN
8893
8894  043442  006002                            ROR     R2
```

```
8895  043444  006001                          ROR     R1
8896  043446  053700  043242                  BIS     @#P3,R0
8897  043452  053700  043244                  BIS     @#P12,R0
8898  043456  042702  120020                  BIC     #PIE1!PIE3!PIE12,R2
8899  043462  050002                          BIS     R0,R2
8900
8901  043464  005000                          CLR     R0
8902  043466  053700  043246                  BIS     @#P22,R0
8903  043472  053700  043250                  BIS     @#P24,R0
8904  043476  042701  002400                  BIC     #PIE22!PIE24,R1
8905  043502  050001                          BIS     R0,R1
8906  043504  000404                          BR      12$
8907
8908                                          ;THE PROGRAM COMES HERE IF R0=0
8909                                          ;SO AFTER ROTATE R0 GETS PUT INTO POSITION 1
8910
8911  043506  006002                  10$:    ROR     R2
8912  043510  006001                          ROR     R1
8913  043512  042702  100000                  BIC     #PIE1,R2
8914  043516  010137  043216          12$:    MOV     R1,@#GECC1      ;SAVE ECC1
8915  043522  010237  043220                  MOV     R2,@#GECC2      ;SAVE ECC2
8916  043526  005737  043222                  TST     @#TSECCG        ;IS HARDWARE TO BE CHECKED
8917                                                                  ;IF =1777777 TEST HARDWARE
8918                                                                  ;IF = 0 DO NOT TEST HARDWARE
8919  043532  001432                          BEQ     14$             ;BRANCH IF HARDWARE NOT TO BE CHECKED
8920
8921
8922                                  ;CHECK HARDWARE
8923  043534  032737  000400  177570          BIT     #SW8,@#SWR      ;IS SWITCH 8 SET
8924  043542  001005                          BNE     15$             ;BRANCH IF SW8 IS SET
8925  043544  032737  000100  177570          BIT     #SW6,@#SWR      ;IS SWITCH 6 SET
8926  043552  001401                          BEQ     15$             ;BRANCH IF SW6 IS NOT SET
8927  043554  000421                          BR      14$             ;IF SWITCH 8 IS NOT SET AND
8928                                                                  ;SWITCH 6 IS SET THEN
8929                                                                  ;DO NOT DO COMPARES
8930  043556  010146                  15$:    MOV     R1,-(SP)        ;GOOD PATTERN REGISTER
8931  043560  042716  174000                  BIC     #174000,(SP)    ;GET ONLY PATTERN BITS
8932  043564  022677  136074                  CMP     (SP)+,@RHEC2    ;COMPARE PATTERN REGISTER
8933  043570  001404                          BEQ     13$             ;BRANCH IF GOOD
8934                                          ;TO SAVE TIME
8935  043572  004737  040126                  JSR     PC,@#PUTREG     ;SAVE REGISTERS
8936  043576  104035                          ERROR   35              ;PATTERN REGISTER IN 11 BITS IN ERROR
8937  043600  000407                          BR      14$             ;BRANCH OUT
8938  043602  023777  043230  136052  13$:    CMP     @#POSITI,@RHEC1 ;COMPARE POISTION REGISTER
8939  043610  001403                          BEQ     14$             ;BRANCH IF GOOD
8940                                          ;TO SAVE TIME
8941  043612  004737  040126                  JSR     PC,@#PUTREG     ;SAVE REGISTERS
8942  043616  104035                          ERROR   35              ;POSITION REGISTER IN ERROR
8943                                                                  ;"DATA ENVLOP" GIVES NUMBER OF CLOCK
8944                                                                  ;PULSES FROM BEGINING OF COMMAND
8945                                                                  ;THAT IS THE CLOCKS IN THE R/W DATA FIELD ENVELOPE
8946                                                                  ;
8947                                                                  ;IN A WRITE THERE ARE 10000 OCTAL CLOCKS
8948                                                                  ;IN A READ THERE ARE 10040 OCTAL CLOCKS
```

```
8949                                                    ;
8950                                                    ;
8951                                                    ;"N-CODE ZEROS" GIVE THE NUMBER OF CLOCKS
8952                                                    ; GIVEN FOR THE LEADING ZEROS FIELD
8953                                                    ;MAX COUNT IS 113713 OCTAL
8954                                                    ;
8955                                                    ;"GOOD POSITION" GIVES NUMBER OF CLOCKS
8956                                                    ;GIVEN AFTER LEADING ZEROS WHICH IS FOR THE DATA
8957                                                    ;FIELD
8958                                                    ;MAX COUNT IS 10040 OR 10041 OCTAL
8959
8960
8961  043620                          14$:
8962  043620  012605                      MOV     (SP)+,R5        ;;POP STACK INTO R5
8963  043622  012604                      MOV     (SP)+,R4        ;;POP STACK INTO R4
8964  043624  012603                      MOV     (SP)+,R3        ;;POP STACK INTO R3
8965  043626  012602                      MOV     (SP)+,R2        ;;POP STACK INTO R2
8966  043630  012601                      MOV     (SP)+,R1        ;;POP STACK INTO R1
8967  043632  012600                      MOV     (SP)+,R0        ;;POP STACK INTO R0
8968  043634  000207                      RTS     PC
```

```
8969
8970
8971                                    ;THIS SUBROUTINE WILL CONTROL THE ECC GENERATION ROUTINE
8972                                    ;FOR ERROR CORRECTION PROCESS
8973                                    ;CALL   JSR,    PC,@#ECORR
8974                                    ;       XP      ;EXPECTED POSITION REGISTER WHEN CORRECTION IS COMPLETE
8975
8976
8977
8978  043636  000000                    ERPOS:  0                       ;POSITION REG. WHEN CORRECTION IS COMPLETE
8979
8980
8981
8982  043640  010037  002002            ECORR:  MOV     R0,@#PCJSR      ;SAVE PC OF JSR + 4
8983  043644  162737  000004  002002            SUB     #4,@#PCJSR      ;SAVE PC OF JSR
8984  043652  012037  043636                    MOV     (R0)+,@#ERPOS   ;GET POSITION REG. WHEN CORRECTION IS COMPLETE
8985  043656  010146                            MOV     R1,-(SP)        ;;PUSH R1 ON STACK
8986  043660  013701  001652                    MOV     @#RHMR,R1       ;MAINTENANCE REGISTER
8987  043664  012711  000001                    MOV     #DMD,@R1        ;SET DIAGNOSTIC MODE BIT
8988  043670  005037  043214                    CLR     @#ECDATA        ;ECC DATA IS ZERO
8989
8990
8991
8992  043674  005737  043230            1$:     TST     @#POSITI        ;IS  SOFTWARE POSITION NON ZERO
8993  043700  001007                            BNE     2$              ;BRANCH IF N-CODE S COMPLETE
8994  043702  005337  043226                    DEC     @#NCOUNT        ;DECREMENT N-CODE
8995  043706  001001                            BNE     6$              ;BRANCH IF N-CODE IS NOT COMPLETE
8996  043710  000403                            BR      2$              ;BRANCH  AS N-CODE IS COMPLETE
8997  043712  005237  043236            6$:     INC     @#ZCODE         ;INCREMENT CLOCKS GIVEN FOR LEADING ZEROS
8998  043716  000420                            BR      3$              ;BRANCH AS N-CODE IS NOT COMPLETE
8999                                                                    ;GO TO GIVE CLOCK AND TEST ECC
9000  043720  005237  043230            2$:     INC     @#POSITI        ;INCREMENT SOFTWARE POSITION
9001  043724  023737  043636  043230            CMP     @#ERPOS,@#POSITI;HAVE ENOUGH CLOCKS BEEN GIVEN TO DETECT ERROR
9002  043732  103012                            BHIS    3$              ;BRANCH IF MORE CLOCKS TO BE GIVEN
9003  043734  023737  043240  043230            CMP     @#HADTMP,@#POSITI;HAVE ENOUGH CLOCKS BEEN GIVEN FOR HARD ERROR
9004                                                                    ;THAT IS HAVE 4128 MORE CLOCKS BEEN GIVEN
9005  043742  001415                            BEQ     5$              ;BRANCH IF YES
9006  043744  032711  000400                    BIT     #ZER,@R1        ;CHECK ZERO DETECT BIT IN RHMR
9007  043750  001016                            BNE     4$              ;BRANCH IS ZER SET
9008                                            ;TO SAVE TIME
9009  043752  004737  040126                    JSR     PC,@#PUTREG     ;SAVE REGISTERS
9010  043756  104034                            ERROR   34              ;ZERO DETECT BIT NOT HIGH
9011                                                                    ;WHEN 21 BITS IN ECC 32 BIT REGISTER IS  0
9012
9013
9014  043760  052711  000002            3$:     BIS     #MCLK,@R1       ;SET CLOCK
9015  043764  042711  000002                    BIC     #MCLK,@R1       ;CLEAR CLOCK
9016  043770  004737  043252                    JSR     PC,@#ECTEST     ;GO TO GENERATE AND TEST ECC
9017  043774  000737                            BR      1$              ;CONTINUE
9018
9019                                    ;THIS EXTRA CLOCK IS TO BRING ECH HIGH
9020                                    ;AFTER THIS CLOCK POSITION REGISTER MAY BE 10040 OR 10041 OCTAL
9021
9022  043776  052711  000002            5$:     BIS     #MCLK,@R1       ;SET CLOCK
```

```
9023  044002  042711  000002                  BIC     #MCLK,@R1       ;CLEAR CLOCK
9024
9025  044006                          4$:
9026  044006  012601                          MOV     (SP)+,R1        ;;POP STACK INTO R1
9027  044010  000200                          RTS     R0
9028
9029
9030
9031
9032
9033
9034                                  ;THIS SUBROUTINE GENERATES THE ECC FOR WHAT IS ON DISK AND INSERTS THEM
9035                                  ;ON LOCATIONS "DISK+1000" AND "DISK+1002"
9036
9037
9038
9039  044012                          FILLEC:
9040  044012  010046                          MOV     R0,-(SP)        ;;PUSH R0 ON STACK
9041  044014  010146                          MOV     R1,-(SP)        ;;PUSH R1 ON STACK
9042  044016  010246                          MOV     R2,-(SP)        ;;PUSH R2 ON STACK
9043  044020  010346                          MOV     R3,-(SP)        ;;PUSH R3 ON STACK
9044  044022  010446                          MOV     R4,-(SP)        ;;PUSH R4 ON STACK
9045  044024  010546                          MOV     R5,-(SP)        ;;PUSH R5 ON STACK
9046  044026  005037  043230                  CLR     @#POSITI        ;CLEAR POSITION
9047  044032  005037  043216                  CLR     @#GECC1         ;CLEAR GECC1
9048  044036  005037  043220                  CLR     @#GECC2         ;CLEAR
9049  044042  012701  047070                  MOV     #DISK,R1        ;POINTER TO DATA FOR ECC GENERATION
9050  044046  012702  000400                  MOV     #256.,R2        ;COUNTER FOR NUMBER OF DATA WORDS
9051  044052  012703  000020          9$:     MOV     #16.,R3         ;COUNTER FOR NUMBER OF BITS PER WORD
9052  044056  012104                          MOV     (R1)+,R4        ;DATA IN R4
9053  044060  006004                  10$:    ROR     R4              ;GET ONE DATA BIT IN CARRY
9054  044062  103004                          BCC     11$             ;BRANCH IF DATA BIT IS ZERO
9055  044064  012737  177777  043214          MOV     #-1,@#ECDATA    ;ECC DATA BIT IS A ONE
9056  044072  000402                          BR      12$             ;BRANCH TO GENERATE ECC
9057  044074  005037  043214          11$:    CLR     @#ECDATA        ;ECC DATA BIT IS A ZERO
9058  044100  004737  043252          12$:    JSR     PC,@#ECTEST     ;GO TO GENERATE ECC
9059  044104  005303                          DEC     R3              ;DECREMENT BIT COUNT
9060  044106  001364                          BNE     10$             ;BRANCH IF 16 BITS NOT DONE
9061  044110  005302                          DEC     R2              ;DECREMENT WORD COUNT
9062  044112  001357                          BNE     9$              ;BRANCH IF 256 WORDS NOT DONE
9063  044114  013737  043216  050070          MOV     @#GECC1,@#DISK+<256.*2>;INSERT ECC1 ON DISK
9064  044122  013737  043220  050072          MOV     @#GECC2,@#DISK+<257.*2>;INSERT ECC2 ON DISK
9065  044130  012605                          MOV     (SP)+,R5        ;;POP STACK INTO R5
9066  044132  012604                          MOV     (SP)+,R4        ;;POP STACK INTO R4
9067  044134  012603                          MOV     (SP)+,R3        ;;POP STACK INTO R3
9068  044136  012602                          MOV     (SP)+,R2        ;;POP STACK INTO R2
9069  044140  012601                          MOV     (SP)+,R1        ;;POP STACK INTO R1
9070  044142  012600                          MOV     (SP)+,R0        ;;POP STACK INTO R0
9071  044144  000207                          RTS     PC
9072
9073
9074
9075
9076
```

```
 9077                                                   .SBTTL  RH BASE ADDRESS CHANGE ROUTINE
 9078
 9079                                           ;*      THIS ROUTINE WILL ALLOW THE CHANGE OF THE BASE
 9080                                           ;*      ADDRESS FROM 176700 TO ANY TYPED VALUE
 9081
 9082  044146                                   BASECH:
 9083  044146  104400  044154                          TYPE    ,.+4            ;;TYPE ASCIZ STRING
 9084  044152  000425                                  BR      64$             ;;GET OVER THE ASCIZ
 9085                                                  ;;.ASCIZ        <15><12>/PRESENT BASE ADDRESS OF REGISTERS IS   /
 9086  044226                                   64$:
 9087  044226  013746  001632                          MOV     @#RHCS1,-(SP)   ;GET READY TO TYPE OLD BASE
 9088  044232  104402                                  TYPOC
 9089  044234  104400  044242                          TYPE    ,.+4            ;;TYPE ASCIZ STRING
 9090  044240  000425                                  BR      65$             ;;GET OVER THE ASCIZ
 9091                                                  ;;.ASCIZ        <15><12>/TYPE NEW BASE ADDRESS FOLLOWED BY 'CR' /
 9092  044314                                   65$:
 9093  044314  004737  052512                          JSR     PC,@#$TKINT     ;INITIALIZE THE TTY KEYBOARD
 9094  044320  104416                                  RDOCT
 9095  044322  012700  001622                          MOV     #RHDB,R0        ;GET STARTING ADDRESS OF REGISTERS
 9096  044326  012701  000024                          MOV     #20.,R1         ;NUMBER OF REGISTERS
 9097  044332  042710  177700          1$:     BIC     #^C77,(R0)      ;CLEAR OLD BASE
 9098  044336  051620                          BIS     (SP),(R0)+      ;SET NEW BASE
 9099  044340  005301                          DEC     R1              ;COUNT
 9100  044342  001373                          BNE     1$              ;BRANCH IF 20 NOT DONE
 9101  044344  104400  044352                  TYPE    ,.+4            ;;TYPE ASCIZ STRING
 9102  044350  000417                          BR      66$             ;;GET OVER THE ASCIZ
 9103                                          ;;.ASCIZ        <15><12>/PRESENT VECTOR ADDRESS IS /
 9104  044410                           66$:
 9105  044410  013746  001620                  MOV     @#RPVEC,-(SP)   ;GET READY TO TYPE OLD VECTOR ADDRESS
 9106  044414  104402                          TYPOC
 9107  044416  104400  044424                  TYPE    ,.+4            ;;TYPE ASCIZ STRING
 9108  044422  000437                          BR      67$             ;;GET OVER THE ASCIZ
 9109                                          ;;.ASCIZ        <15><12>/TYPE NEW VECTOR ADDRESS OR RETYPE OLD ONE FOLLOWED BY "CR
 9110  044522                           67$:
 9111  044522  104416                          RDOCT
 9112  044524  012637  001620                  MOV     (SP)+,@#RPVEC   ;SETUP VECTOR ADDRESS
 9113  044530  104400  044536                  TYPE    ,.+4            ;;TYPE ASCIZ STRING
 9114  044534  000402                          BR      68$             ;;GET OVER THE ASCIZ
 9115                                          ;;.ASCIZ        <15><12>/ /
 9116  044542                           68$:
 9117  044542  104400  044550                  TYPE    ,.+4            ;;TYPE ASCIZ STRING
 9118  044546  000421                          BR      69$             ;;GET OVER THE ASCIZ
 9119                                          ;;.ASCIZ        <15><12>/RESTART PROGRAM FROM 200 OR 210/
 9120  044612                           69$:
 9121  044612  104400  044620                  TYPE    ,.+4            ;;TYPE ASCIZ STRING
 9122  044616  000416                          BR      70$             ;;GET OVER THE ASCIZ
 9123                                          ;;.ASCIZ        <15><12>/NEW BASE WILL REMAIN   - /
 9124  044654                           70$:
 9125  044654  013746  001632                  MOV     @#RHCS1,-(SP)
 9126  044660  104402                          TYPOC
 9127  044662  104400  044670                  TYPE    ,.+4            ;;TYPE ASCIZ STRING
 9128  044666  000416                          BR      71$             ;;GET OVER THE ASCIZ
 9129                                          ;;.ASCIZ        <15><12>/NEW VECTOR WILL REMAIN - /
 9130  044724                           71$:
```

```
 9131  044724  013746  001620                MOV     @#RPVEC,-(SP)
 9132  044730  104402                        TYPOC
 9133  044732  104400  044740                TYPE    .+4             ;;TYPE ASCIZ STRING
 9134  044736  000416                        BR      72$             ;;GET OVER THE ASCIZ
 9135                                        ;;.ASCIZ        <15><12>/UNTIL PROGRAM IS RELOADED/
 9136  044774                        72$:
 9137  044774  000000                        HALT
 9138
 9139
 9140
 9141
 9142
 9143                                ;THIS IS A LITTLE ROUTINE THAT TESTS NED BIT 11 IN RHCS2
 9144                                ;THIS LOOPS HERE FOR EVER
 9145                                ;TO BE USED ONLY IF DRIVES PRESENT LOOKING AT NED DOES NOT AGREE
 9146                                ;WITH WHAT IS REALY THERE
 9147  044776  000000                ERUNIT: 0                       ;UNIT UNDER MANUAL TEST
 9148  045000  004737  040522        ERSTART:JSR     PC,@#CLDISK     ;SET GENERAL REG.
 9149  045004  013712  044776                MOV     @#ERUNIT,@R2    ;SELECT UNIT
 9150  045010  005714                1$:     TST     @R4             ;TEST RHER1
 9151  045012  032712  010000                BIT     #NED,@R2        ;TEST NED
 9152  045016  001401                        BEQ     2$              ;BRANCH IF GOOD
 9153  045020  000773                        BR      1$              ;NED NOT SET
 9154  045022  000772                2$:     BR      1$              ;NED SET
```

```
                        .SBTTL  DISK SIMULATION
                ;*****************************************************************
                ;*****************************************************************
                ;IN A WRITE HEADER AND DATA COMMAND FILL THE FOLLOWING
                ;WCLY=WITH CYLINDER TO BE ON DISK
                ;WSECTR=WITH SECTOR AND TRACK TO BE ON DISK
                ;WKEY1= WITH KEY1 TO BE ON DISK
                ;WKEY2= WITH KEY2 TO BE ON DISK
                ;FNWORD= NO OF DATA WORDS TO BE WRITTEN ON DISK
                ;THE COMMAND THEN IS   JSR PC,COMWHD
                ;
                ;
                ;
                ;IN A WRITE DATA COMMAND FILL THE FOLLOWING
                ;CYL=WITH CYLINDER TO BE FOUND ON DISK
                ;SECOTR= WITH SECTOR AND TRACK TO BE FOUND ON DISK
                ;KEY1= WITH KEY1 TO BE FOUND ON DISK
                ;KEY2= WITH KEY2 TO BE FOUND ON DISK
                ;X= 1  MUST BE ONE
                ;NOWORD= WITH NUMBER OF DATA WORDS TO BE WRITTEN
                ;THE COMMAND THEN IS JSR PC,COMHD
                ;
                ;
                ;
                ;
                ;
                ;IN A READ HEADER AND DATA COMMAND FILL THE FOLLOWING
                ;CYL= WITH CYLINDER TO BE FOUND ON DISK
                ;SECOTR= WITH SECTOR AND TRACK TO BE FOUND ON DISK
                ;KEY1= WITH KEY1 TO BE FOUND ON DISK
                ;KEY2=WITH KEY2 TO BE FOUND ON DISK
                ;DAWORD= WITH NUMBER OF WORDS TO BE FOUND ON DISK
                ;X=0 MUST BE ZERO
                ;THE COMMAND THEN IS JSR PC,COMHD
                ;
                ;
                ;
                ;
                ;
                ;
                ;
                ;
                ;IN A READ DATA COMMAND FILL THE FOLLOWING
                ;CYL= WITH CYLINDER TO BE FOUND ON DISK
                ;SECOTR= WITH SECTOR AND TRACK TO BE FOUND ON DISK
                ;KEY1= WITH KEY1 TO BE FOUND ON DISK
                ;KEY2=WITH KEY2 TO BE FOUND ON DISK
                ;DAWORD= WITH NUMBER OF WORDS TO BE FOUND ON DISK
                ;;X=0 MUST BE ZERO
```

```
9209                                    ;THE COMMAND THEN IS JSR PC,COMHD
9210                                    ;
```

```
9211
9212
9213
9214
9215
9216
9217                                    ;*****************************************************************
9218                                    ;WRITE DATA COMMAND
9219                                    ;OR READ COMMAND I.E DATA ONLY OR HEADER AND DATA
9220                                    ;*****************************************************************
9221
9222  045024  000000                    RUNCTR: .WORD   0
9223  045026  011637  002002            COMHD:  MOV     (SP),@#PCJSR    ;SAVE PC OF JSR + 4
9224  045032  162737  000004  002002            SUB     #4,@#PCJSR      ;SAVE PC OF JSR
9225  045040  010046                            MOV     R0,-(SP)        ;;PUSH R0 ON STACK
9226  045042  010146                            MOV     R1,-(SP)        ;;PUSH R1 ON STACK
9227  045044  010246                            MOV     R2,-(SP)        ;;PUSH R2 ON STACK
9228  045046  010346                            MOV     R3,-(SP)        ;;PUSH R3 ON STACK
9229  045050  010446                            MOV     R4,-(SP)        ;;PUSH R4 ON STACK
9230  045052  010546                            MOV     R5,-(SP)        ;;PUSH R5 ON STACK
9231  045054  012777  000001  134570            MOV     #DMD,@RHMR      ;SET DIAGNOSTIC MODE
9232  045062  052777  000001  134542            BIS     #GO,@RHCS1      ;ISSUE 'GO' BIT & STALL 'TILL 'RUN' LINE
9233                                                                    ;FUNCTION CODE IS ISSUED BY THE TEST
9234
9235  045070  012737  000113  045024  RUNWAT: MOV     #75.,@#RUNCTR   ;LOAD STALL COUNT = APPROX. 450US
9236                                                                    ;FOR 11/50 CPU WITH CORE MEMORY
9237  045076  005337  045024          1$:     DEC     @#RUNCTR        ;COUNT DOWN ONE
9238  045102  001375                          BNE     1$              ;CONTINUE UNTIL = 0
9239
9240  045104  016746  000044                  MOV     SECOTR, -(SP)   ;GET DESIRED SECTOR/TRACK
9241  045110  042716  177740                  BIC     #177740, (SP)   ;MAKE ONLY SECTOR
9242  045114  012637  045124                  MOV     (SP)+, @#TRK    ;SAVE SECTOR
9243  045120  004137  051204                  JSR     R1,@#SEARCH     ;DO SEARCH SECTOR & ISSUE SECTOR CLOCKS
9244  045124  000000          TRK:    .WORD   0
9245  045126  012701  000240                  MOV     #+NOP,R1        ;GOING TO MOVE NOPS
9246  045132  010137  045164                  MOV     R1,@#SSYN       ;NOP INTO SSYN
9247  045136  010137  045166                  MOV     R1,@#HEDGAP     ;NOP INTO HEDGAP
9248  045142  010137  045170                  MOV     R1,@#HEDSYN     ;NOP INTO HEDSYN
9249  045146  004137  045274                  JSR     R1,@#RDHEAD
9250
9251  045152  000000          CYL:    .WORD   0               ;CYLINDER ADDRESS
9252  045154  000000          SECOTR: .WORD   0               ;SECTOR/TRACK ADDRESS
9253  045156  000000          KEY1:   .WORD   0               ;KEY1 WORD
9254  045160  000000          KEY2:   .WORD   0               ;KEY2 WORD
9255  045162  000000          X:      .WORD   0               ;X=1 WRITE COMMAND
9256                                                          ;X=0 READ COMMAND
9257
9258  045164  000240          SSYN:   NOP                     ;IF "ERROR 2" INSERTED BY RDHEAD
9259                                                          ;SUBROUTINE THEN THE FIRST SYNC.
9260                                                          ;IS NOT DETECTED.  NO BAD DATA
9261                                                          ;IS GIVEN BECAUSE SYNC=144000
9262                                                          ;CANNOT BE READ.  WORD NO
9263                                                          ;IS "1" BECAUSE THIS IS THE FIRST
9264                                                          ;WORD TESTED
```

```
 9265
 9266  045166  000240                  HEDGAP: NOP                     ;IF "ERROR 3" INSERTED BY
 9267                                                                  ;RDHEAD SUBROUTINE THEN THE
 9268                                                                  ;HEADER GAP 0'S WERE NOT
 9269                                                                  ;WRITTEN RIGHT.
 9270                                                                  ;IF "WORD NO" CONTAINS, SAY
 9271                                                                  ;3(8), THEN IT IS THE THIRD
 9272                                                                  ;WORD OF A 5 WORD HEADER
 9273                                                                  ;GAP THAT IS WRONG
 9274                                                                  ;"BAD DATA" CONTAINS WHAT IS
 9275                                                                  ;GOING ON THE DISK
 9276
 9277  045170  000240                  HEDSYN: NOP                     ;IF "ERROR 3" INSERTED BY RDHEAD
 9278                                                                  ;SUBROUTINE THEN THE HEADER SYNC.
 9279                                                                  ;GENERATED BY DCL IS WRONG
 9280                                                                  ;OR THE LAST BYTE
 9281                                                                  ;OF THE HEADER GAP 0'S IS WRONG
 9282                                                                  ;IN EITHER CASE WORD NO=6
 9283                                                                  ;RIGHT BYTE IS HEADER 0
 9284                                                                  ;LEFT BYTE IS SYNC
 9285                                                                  ;"BAD DATA" HAS WHAT IS GOING
 9286                                                                  ;ON DISK
 9287
 9288  045172  005737  001774              TST     @#ERFLGS            ;ARE ANY ERRORS DETECTED
 9289  045176  001017                      BNE     OUT                 ;IF YES BRANCH
 9290  045200  005737  045162              TST     @#X
 9291  045204  001410                      BEQ     DAREAD
 9292  045206  005737  045266              TST     @#NOSYNC            ;IS THIS FORCED HEADER ERROR COMMAND
 9293                                                                  ;IF YES  NOSYNC=-1 THEN WRITE OR READ
 9294                                                                  ;IS SHUT OFF SO BRANCH OUT
 9295                                                                  ;IF  NOSYNC=0 THEN CONTINUE
 9296  045212  001011                      BNE     OUT                 ;BRANCH IF SET
 9297  045214  004137  046540              JSR     R1,@#WRDATA         ;WRITE DATA
 9298
 9299  045220  000000          NOWORD: .WORD   0                       ;NO OF WORDS TO BE WRITTEN
 9300  045222  000000          Y:      .WORD   0                       ;
 9301  045224  000404                  BR      OUT                     ;
 9302  045226  004137  051460  DAREAD: JSR     R1,@#REDATA             ;READ DATA
 9303  045232  000000          DAWORD: .WORD   0                       ;NO OF WORDS TO BE READ
 9304  045234  000000                  .WORD   0
 9305  045236                  OUT:
 9306  045236  012605                  MOV     (SP)+,R5                ;;POP STACK INTO R5
 9307  045240  012604                  MOV     (SP)+,R4                ;;POP STACK INTO R4
 9308  045242  012603                  MOV     (SP)+,R3                ;;POP STACK INTO R3
 9309  045244  012602                  MOV     (SP)+,R2                ;;POP STACK INTO R2
 9310  045246  012601                  MOV     (SP)+,R1                ;;POP STACK INTO R1
 9311  045250  012600                  MOV     (SP)+,R0                ;;POP STACK INTO R0
 9312  045252  000207                  RTS     PC                      ;EXIT ROUTINE
```

```
 9313
 9314                                   ;*****************************************************************
 9315                                   ;THE DISK SECTOR IS DEVIDED AS FOLLOWS
 9316                                   ;19 WORDS OF 0, ONE WORD 144000
 9317                                   ;THESE MAKE 39 BYTES FOR SECTOR GAP AND ONE SYNC. BYTE
 9318
 9319
 9320  045254  014400                   RSYNC:  14400
 9321  045256  000000                   RCYL:   0
 9322  045260  000000                   RSETR:  0
 9323  045262  000000                   RKEY1:  0
 9324  045264  000000                   RKEY2:  0
 9325
 9326
 9327                                   ;5 WORDS OF 0 ONE WORD 144000
 9328                                   ;THESE MAKE 11 BYTES FOR HEADER GAP AND ONE SYNC. BYTE
 9329                                   ;THESE ARE DCL GENERATED
 9330
 9331
 9332
 9333                                   ;THERE ARE 256 WORDS OF DATA
 9334                                   ;THERE ARE 2 WORDS FOR ECC GENERATED BY DCL
 9335                                   ;15 WORDS OF 0 FOR DATA GAP AND TOLERANCE GAP
 9336                                   ;*****************************************************************
 9337
 9338
```

```
                                ;*****************************************************************
                                ;READ DISK HEADER
                                ;*****************************************************************



045266  000000                  NOSYNC: 0                       ;FORCED HEADER ERROR = -1
                                                                ;NORMAL = 0
045270  000000                  TY:     0                       ;ERROR TYPE NO.
045272  000000                  ERWORD: 0                       ;ERROR WORD NO.



045274  012137  045256          RDHEAD: MOV     (R1)+, @#RCYL   ;STORE CYLINDER ADDRESS
045300  012137  045260                  MOV     (R1)+, @#RSETR  ;STORE SECTOR AND TRACK ADDRESS
045304  012137  045262                  MOV     (R1)+, @#RKEY1  ;STORE KEY1
045310  012137  045264                  MOV     (R1)+, @#RKEY2  ;STORE KEY2
045314  012137  046064                  MOV     (R1)+, @#COMPA  ;STORE COMPARE OR NOT
045320  010146                          MOV     R1,-(SP)        ;;PUSH R1 ON STACK
045322  013700  001652                  MOV     @#RHMR, R0      ;R0 CONTAINS MAINTANENCE REG.
045326  012705  000002                  MOV     #2,     R5      ;R5 IS A COUNTER FOR WORDS
045332  012710  000001                  MOV     #DMD,   @R0     ;DIAG. MODE
045336  052710  000010                  BIS     #MSTCK,@R0      ;SET SECTOR FOR FIRST WORD
045342  052710  000002                  BIS     #MCLK,@R0       ;SET CLOCK FOR FIRST WORD
045346  042710  000012                  BIC     #MSTCK!MCLK,@R0 ;RESET SECTOR AND CLOCK
045352  000404                          BR      2$              ;BRANCH OVER GIVING SECTOR FOR FIRST TIME
045354  012710  000013          1$:     MOV     #MSTCK!MCLK!DMD,@R0;SET SECTOR, CLOCK, DIAG. MODE, RESET INDEX
045360  042710  000012                  BIC     #MSTCK!MCLK,@R0 ;RESET SECTOR, CLOCK
045364  012702  000007          2$:     MOV     #7,     R2      ;R2 IS A COUNTER FOR BYTES
045370  052710  000002          3$:     BIS     #MCLK,  @R0     ;SET CLOCK
045374  042710  000002                  BIC     #MCLK,  @R0     ;RESET CLOCK
045400  005302                          DEC     R2              ;BYTE COUNTER
045402  001372                          BNE     3$              ;BRANCH IF BYTE NOT COMPLETE
045404  005305                          DEC     R5              ;WORD COUNTER
045406  001362                          BNE     1$              ;BRANCH IF WORD NOT COMPLETE
045410  012702  000022                  MOV     #18.,   R2      ;NO OF WORDS OF ZEROS
045414  005037  046062          4$:     CLR     @#WORD          ;READ 0
045420  004737  046066                  JSR     PC,     @#READ  ;GO TO READ
045424  005302                          DEC     R2              ;COUNT
045426  001372                          BNE     4$
045430  013737  045254  046062        MOV     @#RSYNC,@#WORD  ;SYNC. WORD
045436  004737  046066                  JSR     PC,     @#READ
045442  032710  001000                  BIT     #DTSY,  @R0     ;SYNC. BYTE DETECTED?
045446  001012                          BNE     5$              ;BRANCH IF SYNC DETECTED
045450  012737  000001  045272        MOV     #1,     @#ERWORD        ;ERROR WORD NO
045456  013737  045254  001124        MOV     @#RSYNC,@#$GDDAT        ;SYNC WORD
045464  012737  104002  045164        MOV     #104002,@#SSYN  ;INSERT "ERROR 2" IN SSYN
```

```
 9393  045472  000571                          BR      13$             ;BRANCH OUT
 9394  045474  013737  045256  046062  5$:     MOV     @#RCYL, @#WORD  ;SETUP CYLINDER
 9395  045502  004737  046066                  JSR     PC,     @#READ  ;READ
 9396  045506  013737  045260  046062          MOV     @#RSETR,@#WORD  ;SETUP SECTOR/TRACK
 9397  045514  004737  046066                  JSR     PC,@#READ       ;READ
 9398  045520  013737  045262  046062          MOV     @#RKEY1,@#WORD  ;SETUP KEY1
 9399  045526  004737  046066                  JSR     PC,@#READ       ;READ
 9400  045532  013737  045264  046062          MOV     @#RKEY2,@#WORD  ;SETUP KEY2
 9401  045540  004737  046066                  JSR     PC,@#READ       ;READ
 9402  045544  013737  047052  046062          MOV     @#CRC,@#WORD    ;SETUP CRC
 9403  045552  004737  046066                  JSR     PC,@#READ       ;READ
 9404  045556  005737  002014                  TST     @#TESDTE        ;IS THIS A DRIVE TIMING ERROR
 9405  045562  001135                          BNE     13$             ;BRANCH OUT IF YES
 9406  045564  005737  046064                  TST     @#COMPA         ;IS THIS A READ OR WRITE COMMAND
 9407  045570  001472                          BEQ     11$
 9408  045572  012705  047054                  MOV     #HEGAP, R5      ;POINTER FOR HEADER GAP
 9409  045576  012702  000005                  MOV     #5,     R2      ;NO OF WORDS OF ZEROS
 9410  045602  012737  000006  045272  6$:     MOV     #6,@#ERWORD     ;ERROR WORD NO SET
 9411  045610  004737  046320                  JSR     PC,@#WRITE      ;FOR HEADER GAP
 9412  045614  005737  046316                  TST     @#WWORD         ;TEST WRITTEN WORD
 9413  045620  001413                          BEQ     7$              ;BRANCH IF GOOD THAT IS 0
 9414  045622  160237  045272                  SUB     R2,@#ERWORD     ;WORD NO IN ERROR
 9415  045626  005037  001124                  CLR     @#$GDDAT                ;GOOD WORD SHOULD BE 0
 9416  045632  013767  046316  133266          MOV     @#WWORD,        $BDDAT  ;BAD DATA
 9417  045640  012737  104003  045166          MOV     #104003,@#HEDGAP        ;"ERROR 2" GOES IN HEDGAP
 9418  045646  000503                          BR      13$             ;BRANCH OUT
 9419  045650  013725  046316          7$:     MOV     @#WWORD,(R5)+   ;SAVE HEADER GAP
 9420  045654  005302                          DEC     R2
 9421  045656  001351                          BNE     6$
 9422  045660  004737  046320                  JSR     PC,     @#WRITE ;WRITE HEADER (DATA) GAP SYNC
 9423  045664  023737  045254  046316          CMP     @#RSYNC,@#WWORD
 9424  045672  001426                          BEQ     10$
 9425  045674  005737  045266                  TST     @#NOSYNC        ;IS THIS FORCED HEADER ERROR COMMAND
 9426                                                                  ;IF YES  NOSYNC=-1 THEN WRITE OR READ
 9427                                                                  ;IS SHUT OFF SO BRANCH OUT
 9428                                                                  ;IF NO NOSYNC=0 THEN CONTINUE
 9429  045700  001406                          BEQ     14$             ;BRANCH IF TRUE ERROR
 9430  045702  005737  046316                  TST     @#WWORD         ;
 9431  045706  001420                          BEQ     10$             ;BRANCH IF GOOD
 9432  045710  005037  001124                  CLR     @#$GDDAT        ;IT SHOULD BE ZERO
 9433  045714  000403                          BR      15$             ;BRANCH TO TYPE ERROR
 9434  045716  013737  045254  001124  14$:     MOV     @#RSYNC,@#$GDDAT        ;GOOD DATA
 9435  045724  013737  046316  001126  15$:     MOV     @#WWORD,@#$BDDAT        ;BAD DATA
 9436  045732  012737  000006  045272          MOV     #6,     @#ERWORD
 9437  045740  012737  104003  045170          MOV     #104003,@#HEDSYN
 9438  045746  000443                          BR      13$             ;BRANCH OUT
 9439  045750  013725  046316          10$:    MOV     @#WWORD,(R5)+   ;SAVE DATA SYNC.
 9440  045754  000440                          BR      13$
 9441                                  ;READ COMMAND START FROM HERE
 9442  045756  012702  000005          11$:    MOV     #5,     R2
 9443  045762  005067  000074          12$:    CLR     WORD
 9444  045766  004767  000074                  JSR     PC,     READ    ;READ HEADER GAP
 9445  045772  005302                          DEC     R2              ;IS 5 HEADER GAP ZEROS COMPLETE
 9446  045774  001372                          BNE     12$             ;IF NOT BRANCH
```

```
9447  045776  013737  045254  046062          MOV     @#RSYNC,@#WORD  ;SYNC WORD
9448  046004  004767  000056                  JSR     PC,     READ    ;READ HEADER (DATA) SYNC)
9449  046010  005737  045266                  TST     @#NOSYNC
9450  046014  001404                          BEQ     16$             ;IF NOT ERROR COMMAND BRANCH
9451  046016  032710  001000                  BIT     #DTSY,@RO       ;SYNC. DETECTED
9452  046022  001415                          BEQ     13$             ;IF ZERO BRANCH OUT
9453  046024  000403                          BR      17$             ;IF NOT ZERO BRANCH TO ERROR
9454  046026  032710  001000          16$:    BIT     #DTSY,  @RO     ;SYNC. DETECTED?
9455  046032  001011                          BNE     13$             ;BRANCH IF YES
9456  046034  012737  000006  045272  17$:    MOV     #6,@#ERWORD     ;ERROR WORD NO.
9457  046042  013737  045254  001124          MOV     @#RSYNC,@#SGDDAT;SYNC WORD
9458  046050  012737  104002  045170          MOV     #104002,@#HEDSYN
9459  046056                          13$:
9460  046056  012601                          MOV     (SP)+,R1        ;;POP STACK INTO R1
9461  046060  000201                          RTS     R1
9462
9463
```

```
                                ;*****************************************************************
                                ;READ ONE WORD IN "WORD"
                                ;*****************************************************************


046062  000000          WORD:   0
046064  000000          COMPA:  0


046066                  READ:
046066  010246                  MOV     R2,-(SP)        ;;PUSH R2 ON STACK
046070  012705  000002          MOV     #2,     R5      ;WORD COUNTER
046074  012710  000001          MOV     #DMD,   @R0     ;SET DIAG. MODE
046100  006037  046062          ROR     @#WORD          ;CHECKING IF THERE IS A ONE
046104  103002                  BCC     1$              ;IF NO ONE BRANCH
046106  052710  000020          BIS     #MRD,   @R0     ;SET BIT 4 IF DATA HAS ONE
046112  012702  000007  1$:     MOV     #7,     R2      ;BYTE COUNTER
046116  052710  000012          BIS     #MSTCK!MCLK,    @R0     ;SET CLOCK,DATA IF ANY, SECTOR
046122  005737  043222          TST     @#TSECCG        ;IS THIS BIT TO GENERATE AND TEST ECC
046126  001411                  BEQ     6$              ;BRANCH IF NO
046130  032710  000020          BIT     #MRD,@R0        ;IS DATA BIT A ONE
046134  001404                  BEQ     5$              ;BRANCH IF DATA BIT IS 0
046136  012737  177777  043214  MOV     #-1,@#ECDATA    ;ECC DATA BIT IS A ONE
046144  000402                  BR      6$              ;BRANCH
046146  005037  043214  5$:     CLR     @#ECDATA        ;ECC DATA BIT IS A 0
046152  012746  000001  6$:     MOV     #DMD,   -(SP)   ;KEEP ONLY DIAG. MODE
046156  006037  046062          ROR     @#WORD          ;CHECKING IF THERE IS A ONE
046162  103002                  BCC     2$              ;IF NO ONE BRANCH
046164  012716  000021          MOV     #MRD!DMD,       (SP)    ;KEEP DATA AND DIAG. MODE
046170  012610          2$:     MOV     (SP)+,  @R0     ;PUT IN DATA,RESET CLOCK, SECTOR
046172  005737  043222          TST     @#TSECCG        ;IS ECC TO BE GENERATED FOR THIS BIT
046176  001404                  BEQ     3$              ;BRANCH IF NO
046200  005237  043234          INC     @#DATENV        ;NUMBER OF CLOCKS GIVEN FOR DATA ENVELOPE
046204  004737  043252          JSR     PC,@#ECTEST     ;GO TO GENERATE AND TEST ECC
046210  052710  000002  3$:     BIS     #MCLK,  @R0     ;SET CLOCK
046214  005737  043222          TST     @#TSECCG        ;IS THIS BIT TO GENERATE ECC
046220  001411                  BEQ     8$              ;BRANCH IF NO
046222  032710  000020          BIT     #MRD,@R0        ;IS DATA BIT A ONE
046226  001404                  BEQ     7$              ;BRANCH IF DATA BIT IS = 0
046230  012737  177777  043214  MOV     #-1,@#ECDATA    ;ECC DATA BIT IS A ONE
046236  000402                  BR      8$              ;BRANCH
046240  005037  043214  7$:     CLR     @#ECDATA        ;ECC DATA BIT IS = 0
046244  012746  000001  8$:     MOV     #DMD,   -(SP)   ;KEEP DIAG. MODE
046250  006037  046062          ROR     @#WORD          ;CHECKING IF THERE IS A ONE
```

```
9518  046254  103002                        BCC     4$              ;BRANCH IF NO ONE
9519  046256  012716  000021                MOV     #MRD!DMD,(SP)   ;KEEP DIAG. MODE AND DATA
9520  046262  012610                4$:     MOV     (SP)+,@RD       ;SET DATA, DIAG. MODE, CLEAR CLOCK
9521  046264  005737  043222                TST     @#TSECCG        ;IS THIS BIT TO GENERATE ECC
9522  046270  001404                        BEQ     9$              ;BRANCH IF NO
9523  046272  005237  043234                INC     @#DATENV        ;NUMBER OF CLOCKS GIVEN FOR DATA ENVELOPE
9524  046276  004737  043252                JSR     PC,@#ECTEST     ;GO TO GENERATE AND TEST ECC
9525  046302  005302                9$:     DEC     R2              ;BYTE COUNTER
9526  046304  001341                        BNE     3$              ;BRANCH IF ONE BYTE NOT COMPLETE
9527  046306  005305                        DEC     R5              ;WORD COUNTER
9528  046310  001300                        BNE     1$              ;BRANCH IF ONE WORD NOT COMPLETE
9529  046312  012602                        MOV     (SP)+,R2        ;;POP STACK INTO R2
9530  046314  000207                        RTS     PC
9531
9532
```

```
                                ;*****************************************************************
                                ;WRITE ONE WORD WHICH COMES BACK IN "WWORD"
                                ;*****************************************************************



046316  000000                  WWORD:  0


046320                          WRITE:
046320  010046                          MOV     R0,-(SP)        ;;PUSH R0 ON STACK
046322  010246                          MOV     R2,-(SP)        ;;PUSH R2 ON STACK
046324  010346                          MOV     R3,-(SP)        ;;PUSH R3 ON STACK
046326  010546                          MOV     R5,-(SP)        ;;PUSH R5 ON STACK
046330  012705  000002                  MOV     #2,R5           ;WORD COUNTER
046334  012710  000001                  MOV     #1,@R0          ;SET DIAG. MODE
                                                                ;R0 HAS RHMR ADDRESS IN IT
046340  012702  000007          1$:     MOV     #7,R2           ;BYTE COUNTER
046344  012710  000013                  MOV     #MSTCK!MCLK!DMD,@R0;SET SECTOR AND CLOCK
046350  032710  000040                  BIT     #MWR,@R0        ;CHECK WRITE BIT IN MAINT. REG.
046354  001406                          BEQ     2$              ;BRANCH IF ZERO
046356  012737  177777  043214          MOV     #-1,@#ECDATA    ;ECC DATA BIT IS A ONE
046364  000261                          SEC                     ;SET CARRY
046366  006003                          ROR     R3              ;MOVE 1 FORWARD
046370  000404                          BR      3$
046372  005037  043214          2$:     CLR     @#ECDATA        ;ECC DATA BIT IS = 0
046376  000241                          CLC                     ;CLEAR CARRY
046400  006003                          ROR     R3              ;MOVE 0 FOR WWORD
046402  012710  000001          3$:     MOV     #DMD,@R0        ;CLEAR SECTOR AND CLOCK
046406  005737  043222                  TST     @#TSECCG        ;IS THIS BIT TO GENERATE ECC
046412  001404                          BEQ     4$              ;BRANCH IF NO
046414  005237  043234                  INC     @#DATENV        ;NUMBER OF CLOCKS GIVEN FOR DATA ENVELOPE
046420  004737  043252                  JSR     PC,@#ECTEST     ;GO TO GENERATE AND TEST ECC
046424  052710  000002          4$:     BIS     #MCLK,@R0       ;SET CLOCK IN RHMR
046430  032710  000040                  BIT     #MWR,@R0        ;CHECK WRITE BIT IN RHMR
046434  001406                          BEQ     5$              ;BRANCH IF ZERO
046436  012737  177777  043214          MOV     #-1,@#ECDATA    ;ECC DATA BIT IS A ONE
046444  000261                          SEC                     ;SET CARRY
046446  006003                          ROR     R3              ;MOVE 1 FOR WWORD
046450  000404                          BR      6$
046452  005037  043214          5$:     CLR     @#ECDATA        ;ECC DATA BIT IS ZERO
046456  000241                          CLC                     ;CLEAR CARRY
046460  006003                          ROR     R3              ;MOVE 0 FOR WWORD
046462  012710  000001          6$:     MOV     #DMD,   @R0     ;CLEAR CLOCK
046466  005737  043222                  TST     @#TSECCG        ;IS THIS BIT TO GENERATE ECC
```

```
 9587  046472  001404                          BEQ     7$              ;BRANCH IF NO
 9588  046474  005237  043234                  INC     @#DATENV        ;NUMBER OF CLOCKS GIVEN FOR DATA ENVELOPE
 9589  046500  004737  043252                  JSR     PC,@#ECTEST     ;GO TO GENERATE AND TEST ECC
 9590  046504  005302                  7$:     DEC     R2              ;COUNT FOR BYTE END
 9591  046506  001346                          BNE     4$              ;IF NOT BYTE END BRANCH
 9592  046510  005305                          DEC     R5              ;COUNT FOR WORD END
 9593  046512  001312                          BNE     1$              ;IF NOT WORD END BRANCH
 9594
 9595  046514  010337  046316                  MOV     R3,@#WORD       ;STORE THE WORD
 9596
 9597  046520  012605                          MOV     (SP)+,R5        ;;POP STACK INTO R5
 9598  046522  012603                          MOV     (SP)+,R3        ;;POP STACK INTO R3
 9599  046524  012602                          MOV     (SP)+,R2        ;;POP STACK INTO R2
 9600  046526  012600                          MOV     (SP)+,R0        ;;POP STACK INTO R0
 9601  046530  000207                          RTS     PC
 9602
 9603
```

```
                                ;*****************************************************************
                                ;WRITE DATA - PUT DATA INTO "DISK" AREA FROM "WWORD"
                                ;ONE WORD AT A TIME
                                ;*****************************************************************
                                
                                
                                
```

```
 9658  046706  012701  000002            MOV     #2,R1
 9659  046712  004767  177402      5$:   JSR     PC,WRITE          ;WRITE ECC1 AND ECC2 ON SIMULATED DISK
 9660  046716  013723  046316            MOV     @#WWORD,(R3)+     ;STORE ON WEEC1 AND WEEC2
 9661  046722  005301                    DEC     R1
 9662  046724  001372                    BNE     5$
 9663  046726  004767  177366            JSR     PC,WRITE          ;WRITE DATA GAP INTO "WWORD"
 9664  046732  013723  046316            MOV     @#WWORD,(R3)+     ;STORE INTO "DISK"
 9665  046736  012701  000016            MOV     #14.,R1
 9666  046742  004737  046320      6$:   JSR     PC,@#WRITE        ;WRITE TOLERANCE GAP ZEROS
 9667  046746  013723  046316            MOV     @#WWORD,(R3)+     ;STORE INTO "DISK"
 9668  046752  005301                    DEC     R1
 9669  046754  001372                    BNE     6$
 9670
 9671  046756  012604                    MOV     (SP)+,R4          ;;POP STACK INTO R4
 9672  046760  012603                    MOV     (SP)+,R3          ;;POP STACK INTO R3
 9673  046762  012602                    MOV     (SP)+,R2          ;;POP STACK INTO R2
 9674  046764  012601                    MOV     (SP)+,R1          ;;POP STACK INTO R1
 9675  046766  012600                    MOV     (SP)+,R0          ;;POP STACK INTO R0
 9676
 9677  046770  000201                    RTS     R1
 9678
 9679
```

```
                        ;*****************************************************************
                        ;WRITE HEADER AND DATA
                        ;
                        ;
                        ;THIS IS THE SIMULATED DISK
                        ;ONLY ONE SECTOR OF SPACE IS ALLOWED
                        ;*****************************************************************
046772  000023  SECGAP: .BLKW   19.          ;SECTOR GAP 38 BYTES OF 0
047040  000001  WSSYNC: .BLKW   1            ;SECTOR GAP 1 BYTE OF 0 ONE SYNC BYTE
047042  000004  HEADER: .BLKW   4            ;HEADER = CYL, SECTOR/TRACK, KEY1, KEY2
047052  000001  WCRC:   .BLKW   1            ;CRC
047054  000005  HEGAP:  .BLKW   5            ;HEADER GAP 10 BYTES OF 0
047066  000001  HDWSYN: .BLKW   1            ;HEADER GAP 1 BYTE OF 0 ONE SYNC. BYTE
047070          SILOTB:                 ;USED IN SILO TEST AS SILO TABLE
047070  000400  DISK:   .BLKW   256.         ;DATA SPACE
050070  000001  WECC1:  .BLKW   1            ;ECC1
050072  000001  WECC2:  .BLKW   1            ;ECC2
050074  000001  DTAGAP: .BLKW   1            ;DATA GAP 2 BYTES OF 0
050076  000016  TOLGAP: .BLKW   14.          ;TOLERANCE GAP 28 BYTES OF 0
```

```
                                ;*****************************************************************
                                ;WRITE HEADER AND DATA
                                ;*****************************************************************


050132  000000          RNCTR1: .WORD   0               ;'RUN' LINE STALL COUNTER
050134  011637  002002  COMMND: MOV     (SP),@#PCJSR    ;SAVE PC OF JSR + 4
050140  162737  000004  002002  SUB     #4,@#PCJSR      ;SAVE PC OF JSR
050146  010046                  MOV     R0,-(SP)        ;;PUSH R0 ON STACK
050150  010146                  MOV     R1,-(SP)        ;;PUSH R1 ON STACK
050152  010246                  MOV     R2,-(SP)        ;;PUSH R2 ON STACK
050154  010346                  MOV     R3,-(SP)        ;;PUSH R3 ON STACK
050156  010446                  MOV     R4,-(SP)        ;;PUSH R4 ON STACK
050160  010546                  MOV     R5,-(SP)        ;;PUSH R5 ON STACK
050162  012777  000001  131462  MOV     #DMD,@RHMR      ;SET DIAGNOSTIC MODE
050170  052777  000001  131434  BIS     #GO,@RHCS1      ;SET 'GO' BIT & STALL 'TILL 'RUN'

050176  012737  000113  050132  RNWAT1: MOV     #75.,@#RNCTR1   ;LOAD STALL COUNTER = APPROX. 450US
                                                        ;FOR 11/50 CPU WITH CORE MEMORY
050204  005337  050132          1$:     DEC     @#RNCTR1        ;COUNT DOWN 1 TIME
050210  001375                  BNE     1$              ;CONTINUE UNTIL 0

050212  013746  050276                  MOV     @#WSECTR,-(SP)  ;GET DESIRED SECTOR/TRACK
050216  042716  177740                  BIC     #177740,(SP)    ;MAKE ONLY SECTOR
050222  012637  050232                  MOV     (SP)+, @#WTRK   ;SAVE SECTOR
050226  004137  051204                  JSR     R1,@#SEARCH     ;DO SEARCH SECTOR & ISSUE SECTOR CLOCKS
050232  000000          WTRK:   .WORD   0               ;SECTOR NO.

050234  012701  000240                  MOV     #+NOP,R1        ;GOING TO MOVE NOPS
050240  010137  050306                  MOV     R1,@#SEGPER     ;NOP INTO SEGAP
050244  010137  050310                  MOV     R1,@#FSYNER     ;NOP INTP FSYNER
050250  010137  050312                  MOV     R1,@#ERHEAD     ;NOP INTO ERHEAD
050254  010137  050314                  MOV     R1,@#ERCRC      ;NOP INTO ERCRC
050260  010137  050316                  MOV     R1,@#ERHDGP     ;NOP INTO ERHDGAP
050264  010137  050320                  MOV     R1,@#HDESYN     ;NOP INTO HDESYN

050270  004137  050370                  JSR     R1,@#WRHEAD     ;WRITE THE HEADER
050274  000000          WCYL:   0               ;CYLINDER
050276  000000          WSECTR: 0               ;SECTOR AND TRACK
050300  000000          WKEY1:  0               ;KEY1
050302  000000          WKEY2:  0               ;KEY2
050304  000000          GCRC:   0               ;GOOD CRC


050306  000240          SEGPER: NOP             ;IF "ERROR 6" INSERTED BY
                                                ;WRHEAD SUBROUTINE THEN
                                                ;SECTOR GAP GOING ON DISK
                                                ;IS NOT RIGHT.
```

```
                                                        ;WORD NO. CONTAINS WHICH
                                                        ;WORD IS WRONG, THAT IS
                                                        ;FIRST OF TENTH OR WHAT EVER NO.
                                                        ;BAD WORD IS GOING ON DISK

050310  000240                  FSYNER: NOP             ;IF "ERROR 6" INSERTED BY
                                                        ;WRHEAD SUBROUTINE THEN
                                                        ;THE LAST 0 BYTE OF SECTOR
                                                        ;GAP, OR FIRST SYNC. BYTE
                                                        ;AFTER SECTOR GAP IS IN
                                                        ;ERROR.

                                                        ;WORD NO. CONTAINS 20
                                                        ;RIGHT BYTE IS SECTOR GAP
                                                        ;LEFT BYTE IS SYNC. BYTE
                                                        ;BAD WORD IS WHAT IS GOING ON
                                                        ;DISK.

050312  000240                  ERHEAD: NOP             ;IF "ERROR 6" INSERTED BY
                                                        ;WRHEAD SUBROUTINE THEN
                                                        ;HEADER GOING ON DISK
                                                        ;IS WRONG.

                                                        ;WORD NO 1 = CYLINDER NO
                                                        ;WORD NO 2 = SECTOR/TRACK
                                                        ;WORD NO 3 = KEY1
                                                        ;WORD NO 4 = KEY2
                                                        ;BAD WORD IS WHAT IS GOING ON
                                                        ;DISK

050314  000240                  ERCRC:  NOP

                                                        ;IF "ERROR 6" INSERTED BY
                                                        ;WRHEAD SUBROUTINE THEN CRC WRITTEN
                                                        ;ON DISK IS IN ERROR.

                                                        ;GOOD DATA IS WHAT SHOULD BE ON DISK
                                                        ;BAD DATA IS WHAT IS GOING ON DISK
                                                        ;WORD NO IS 5.

050316  000240                  ERHDGF. NOP

                                                        ;IF "ERROR 6" INSERTED BY
                                                        ;WRHEAD SUBROUTINE THEN HEADER
                                                        ;GAP GOING ON DISK IS WRONG.

                                                        ;WORD NO. GIVES WHICH OF
                                                        ;THE HEADER GAP WORDS
                                                        ;ARE WRONG. FOR EXAMPLE:

                                                        ;WORD NO 1 = FIRST HEADER
                                                        ;            GAP WORD
                                                        ;BAD WORD IS WHAT IS GOING ON DISK

050320  000240                  HDESYN: NOP

                                                        ;IF "ERROR 6" INSERTED BY
```

```
 9822                                                          ;WRHEAD SUBROUTINE THEN LAST
 9823                                                          ;HEADER GAP BYTE OR HEADER
 9824                                                          ;SYNC BYTE GOING ON DISK IS WRONG.
 9825
 9826                                                          ;WORD NO = 5
 9827                                                          ;BAD DATA IS WHAT IS GOING
 9828                                                          ;ON DISK, RIGHT BYTE IS HEADER
 9829                                                          ;GAP 0 BYTE. LEFT BYTE IS HEADER
 9830                                                          ;GAP SYNC.
 9831
 9832
 9833  050322  005737  001774                TST     @#ERFLGS        ;ARE ANY ERRORS DETECTED
 9834  050326  001004                        BNE     FOUT            ;IF YES EXIT
 9835
 9836  050330  004137  046540                JSR     R1,@#WRDATA     ;WRITE THE DATA
 9837  050334  000000          FNWORD:       .WORD   0               ;FORMAT COMMAND NO. OF DATA
 9838  050336  000000                        .WORD   0
 9839
 9840  050340                  FOUT:
 9841  050340  012605                        MOV     (SP)+,R5        ;;POP STACK INTO R5
 9842  050342  012604                        MOV     (SP)+,R4        ;;POP STACK INTO R4
 9843  050344  012603                        MOV     (SP)+,R3        ;;POP STACK INTO R3
 9844  050346  012602                        MOV     (SP)+,R2        ;;POP STACK INTO R2
 9845  050350  012601                        MOV     (SP)+,R1        ;;POP STACK INTO R1
 9846  050352  012600                        MOV     (SP)+,R0        ;;POP STACK INTO R0
 9847
 9848  050354  000207                        RTS     PC
```

```
 9849
 9850                                   ;*****************************************************************
 9851                                   ;WRITE HEADER
 9852                                   ;*****************************************************************
 9853
 9854
 9855
 9856
 9857
 9858                                   ;R0 = MAINT.REG.
 9859                                   ;R1 = SIMULATED DISK
 9860                                   ;R2 = BYTE COUNT
 9861                                   ;R3 = WRITE WORD
 9862                                   ;R5 = WORD COUNT
 9863
 9864
 9865
 9866  050356  000000                   SCYL:   0                       ;
 9867  050360  000000                   SSECTR: 0                       ;
 9868  050362  000000                   SKEY1:  0                       ;
 9869  050364  000000                   SKEY2:  0                       ;
 9870  050366  000000                   SCRC:   0                       ;
 9871
 9872
 9873  050370  012137  050356           WRHEAD: MOV     (R1)+,@#SCYL
 9874  050374  012137  050360                   MOV     (R1)+,@#SSECTR
 9875  050400  012137  050362                   MOV     (R1)+,@#SKEY1
 9876  050404  012137  050364                   MOV     (R1)+,@#SKEY2
 9877  050410  012137  050366                   MOV     (R1)+,@#SCRC
 9878  050414  010146                           MOV     R1,-(SP)        ;;PUSH R1 ON STACK
 9879  050416  012701  046772                   MOV     #SECGAP,R1      ;SIMULATED DISK INDICATOR
 9880  050422  013700  001652                   MOV     @#RHMR,R0       ;R0 NOW HAS MAINT. REG. ADDR.
 9881  050426  012710  000001                   MOV     #DMD,@R0        ;SET DIAG. MODE IN RHMR
 9882  050432  012705  000002                   MOV     #2,R5           ;WORD COUNTER
 9883  050436  052710  000010                   BIS     #MSTCK,@R0      ;SET SECTOR FOR FIRST BYTE
 9884  050442  012710  000013           1$:     MOV     #MSTCK!MCLK!DMD,@R0;SET SECTOR, CLOCK, DIAG. MODE, RESET INDEX
 9885  050446  032710  000040                   BIT     #MWR,@R0        ;CHECK WRITE BIT IN MAINT. REG.
 9886  050452  001403                           BEQ     2$
 9887  050454  000261                           SEC                     ;SET CARRY
 9888  050456  006003                           ROR     R3              ;MOVE ONE FORWARD
 9889  050460  000402                           BR      3$              ;
 9890  050462  000241           2$:     CLC                             ;CLEAR CARRY
 9891  050464  006003                           ROR     R3              ;MOVE ZERO FORWARD
 9892  050466  012710  000001   3$:     MOV     #DMD,@R0        ;CLEAR CLOCK, SECTOR
 9893  050472  012702  000007                   MOV     #7,R2           ;BYTE COUNTER
 9894  050476  052710  000002   4$:     BIS     #MCLK,@R0       ;SET CLOCK
 9895  050502  032710  000040                   BIT     #MWR,@R0        ;CHECK WRITE BIT IN MAINT.REG.
 9896  050506  001403                           BEQ     5$              ;BRANCH IF ZERO
 9897  050510  000261                           SEC                     ;SET CARRY
 9898  050512  006003                           ROR     R3              ;MOVE ONE FORWARD
 9899  050514  000402                           BR      6$
 9900  050516  000241           5$:     CLC
 9901  050520  006003                           ROR     R3
 9902  050522  012710  000001   6$:     MOV     #DMD,@R0        ;SET DIAG. MODE AGAIN IN RHMR
```

```
 9903  050526  005302                          DEC     R2
 9904  050530  001362                          BNE     4$
 9905  050532  005305                          DEC     R5
 9906  050534  001342                          BNE     1$              ;CONTINUE
 9907
 9908  050536  010321                          MOV     R3,(R1)+
 9909  050540  005703                          TST     R3
 9910  050542  001414                          BEQ     7$
 9911  050544  012737  000001  045272          MOV     #1,@#ERWORD
 9912  050552  005037  001124                  CLR     @#$GDDAT
 9913  050556  010337  001126                  MOV     R3,@#$BDDAT
 9914  050562  012737  104006  050306          MOV     #104006,@#SEGPER
 9915  050570  000137  051176                  JMP     @#17$           ;BRANCH OUT ---------------------)
 9916
 9917  050574  012702  000022          7$:     MOV     #18.,R2         ;COUNT NO. OF SECTOR GAP
 9918  050600  012737  000024  045272  10$:     MOV     #20.,@#ERWORD   ;COUNT TO GIVE ERROR WORD
 9919  050606  004737  046320                  JSR     PC,@#WRITE      ;WRITE SECTOR GAP
 9920  050612  013721  046316                  MOV     @#WORD,(R1)+    ;STORE SECTOR GAP WORD
 9921  050616  001413                          BEQ     11$
 9922  050620  160237  045272                  SUB     R2,@#ERWORD     ;IF NOT GET ERROR WORD NO.
 9923  050624  005037  001124                  CLR     @#$GDDAT        ;GOOD WORD
 9924  050630  013737  046316  001126          MOV     @#WORD,@#$BDDAT;BAD WORD
 9925  050636  012737  104006  050306          MOV     #104006,@#SEGPER;STORE "ERROR 6" IN SEGPER
 9926  050644  000554                          BR      17$             ;BRANCH OUT ---------------------)
 9927
 9928  050646  005302                  11$:     DEC     R2              ;HAVE 18 WORDS OF ZEROS BEEN WRITTEN ?
 9929  050650  001353                          BNE     10$             ;IF NOT DO SO
 9930
 9931                                          ;AT THIS POINT THE SECTOR FOUND FLOP SHOULD
 9932                                          ;BE HIGH.   SO THAT THE HEADER SYNC BYTE CAN BE GIVEN
 9933
 9934                                          ;HOWEVER IN THE DRIVE TIMING ERROR TEST THE REST OF THE ROUTINE
 9935                                          ;IS ABORTED - HEADER SYNC BYTE IS NOT GIVEN
 9936
 9937  050652  005737  002014                  TST     @#TESDTE        ;IS THIS A DRIVE TIMING ERROR
 9938  050656  001147                          BNE     17$             ;BRANCH OUT IF YES
 9939  050660  004737  046320                  JSR     PC,@#WRITE      ;WRITE ONE SECTOR GAP 0 BYTE
 9940                                                                  ;AND ONE SYNC. BYTE = 230
 9941  050664  013711  046316                  MOV     @#WORD,(R1)     ;SAVE 0 BYTE AND SYNC BYTE
 9942  050670  023721  045254                  CMP     @#RSYNC,(R1)+   ;IF SYNC. BYTE RIGHT
 9943  050674  001414                          BEQ     12$             ;IF YES BRANCH
 9944  050676  012737  000024  045272          MOV     #20.,@#ERWORD   ;IF NOT GET READY FOR ERROR
 9945  050704  013737  045254  001124          MOV     @#RSYNC,@#$GDDAT;GOOD WORD
 9946  050712  014137  001126                  MOV     -(R1),@#$BDDAT  ;BAD WORD
 9947  050716  012737  104006  050310          MOV     #104006,@#FSYNER;INSERT "ERROR 6" IN FSYNER
 9948  050724  000524                          BR      17$             ;BRANCH OUT ---------------------)
 9949
 9950  050726  012702  000004          12$:     MOV     #4,R2           ;FOUR HEADER WORDS
 9951  050732  012703  050356                  MOV     #SCYL,R3        ;POINTER FOR HEADER TABLE
 9952  050736  012737  000005  045272  13$:     MOV     #5,@#ERWORD     ;ERROR WORD NO SET
 9953  050744  004737  046320                  JSR     PC,@#WRITE      ;WRITE 4 HEADER WORDS
 9954  050750  013711  046316                  MOV     @#WORD,(R1)     ;STORE WRITTEN WORD
 9955  050754  022321                          CMP     (R3)+,(R1)+     ;IS IT RIGHT?
 9956  050756  001412                          BEQ     14$             ;IF GOOD CONTINUE
```

```
 9957                                                          ;IF NOT GET READY FOR PRINT
 9958  050760  160237  045272                 SUB     R2,@#ERWORD     ;WORD NO
 9959  050764  014337  001124                 MOV     -(R3),@#$GDDAT  ;GOOD DATA
 9960  050770  014137  001126                 MOV     -(R1),@#$BDDAT  ;BAD DATA
 9961  050774  012737  104006  050312         MOV     #104006,@#ERHEAD;INSERT "ERROR 6"
 9962  051002  000475                 BR      17$             ;BRANCH OUT ------------------------>
 9963
 9964  051004  005302                 14$:    DEC     R2              ;ARE 4 HEADER WORDS DONE?
 9965  051006  001353                 BNE     13$             ;IF NOT DO THEM
 9966  051010  004737  046320                 JSR     PC,@#WRITE      ;WRITE CRC
 9967  051014  013711  046316                 MOV     @#WORD,(R1)     ;STORE CRC
 9968  051020  022137  050304                 CMP     (R1)+,@#GCRC    ;COMPARE GOOD CRC
 9969  051024  001414                 BEQ     20$             ;BRANCH IF GOOD
 9970  051026  014137  001126                 MOV     -(R1),@#$BDDATA ;BAD CRC WRITTEN
 9971  051032  013737  050304  001124         MOV     @#GCRC,@#$GDDAT ;GOOD CRC
 9972  051040  012737  000005  045272         MOV     #5,@#ERWORD     ;ERROR WORD NO
 9973  051046  012737  104006  050314         MOV     #104006,@#ERCRC ;INSERT ERROR 6
 9974  051054  000450                 BR      17$             ;EXIT ------------------------------>
 9975
 9976  051056  012702  000005         20$:    MOV     #5,R2           ;NO OF HEADER GAP
 9977  051062  012737  000006  045272 15$:    MOV     #6,@#ERWORD     ;ERROR WORD NO SET
 9978  051070  004737  046320                 JSR     PC,@#WRITE      ;WRITE HEADER GAP
 9979  051074  013721  046316                 MOV     @#WORD,(R1)+    ;STORE
 9980  051100  001412                 BEQ     16$             ;IF GOOD BRANCH
 9981  051102  160237  045272                 SUB     R2,@#ERWORD     ;ERROR WORD NO
 9982  051106  005037  001124                 CLR     @#$GDDAT        ;GOOD DATA
 9983  051112  014137  001126                 MOV     -(R1),@#$BDDAT  ;BAD DATA
 9984  051116  012737  104006  050316         MOV     #104006,@#ERHDGP;STORE "ERROR 6"
 9985  051124  000424                 BR      17$             ;BRANCH OUT ------------------------>
 9986
 9987  051126  005302                 16$:    DEC     R2              ;ARE 5 HEADER GAP ZEROS  DONE
 9988  051130  001354                 BNE     15$             ;IF NOT BRANCH
 9989  051132  004737  046320                 JSR     PC,@#WRITE
 9990  051136  013711  046316                 MOV     @#WORD,(R1)
 9991  051142  023721  045254                 CMP     @#RSYNC,(R1)+
 9992  051146  001413                 BEQ     17$             ;EXIT ------------------------------>
 9993  051150  012737  000005  045272         MOV     #5,@#ERWORD
 9994  051156  014137  001126                 MOV     -(R1),@#$BDDAT
 9995  051162  013737  045254  001124         MOV     @#RSYNC,@#$GDDAT
 9996  051170  012737  104006  050320         MOV     #104006,@#HDESYN
 9997
 9998  051176                 17$:
 9999  051176  012601                 MOV     (SP)+,R1        ;;POP STACK INTO R1
10000
10001  051200  000201                 RTS     R1
10002
10003
```

```
10004
10005
10006
10007
10008
10009
10010                                   ;*****************************************************************
10011                                   ;SEARCH SECTOR
10012                                   ;*****************************************************************
10013
10014
10015
10016
10017
10018
10019                                   ;       R0=RHMR ADDRESS
10020                                   ;       R1=PASSED ARGUMENT (SECTOR SEARCHED FOR)
10021                                   ;       R2=CLOCK COUNT (PER BYTE)
10022                                   ;       R3=SECTOR COUNTER FROM R1
10023                                   ;       R5=BYTES PER WORD COUNT
10024                                   ;BEBORE INDEX IS GIVEN TWO SECTOR CLOCKS ARE GIVEN TO RESET
10025                                   ;SECTOR PULSE IN CASE IT IS SET
10026                                   ;AT BEGINNING OF EACH SECTOR ONE SECTOR CLOCK HAS TO RISE
10027                                   ;BEFORE CLOCK THEN EVERY EIGHT CLOCKS ONE SECTOR CLOCK IS
10028                                   ;IDENTICAL WITH CLOCK
10029                                   ;NUMBERING THE SECTOR CLOCKS AS FOLLOWS
10030                                   ;THE SECTOR CLOCK UNDER INDEX - 0
10031                                   ;THE NEXT - 1
10032                                   ;THE NEXT  - 2
10033                                   ;ETC.
10034                                   ;THEN THE LAST SECTOR CLOCK IN ONE SECTOR HAS NUMBER - 608
10035                                   ;THE NEXT SECTOR THEN HAS 608 SECTOR CLOCKS
10036                                   ;THE NEXT SECTOR THEN HAS ANOTHER 608 SECTOR CLOCKS
10037                                   ;AND SO ON
10038
10039
10040
10041  051202  000000                  SECTR:  0                       ;SECTOR SEARCHED FOR
10042
10043  051204  012137  051202          SEARCH: MOV     (R1)+, @#SECTR  ;SAVE SECTOR SEARCHED FOR
10044  051210  010046                          MOV     R0,-(SP)        ;;PUSH R0 ON STACK
10045  051212  010146                          MOV     R1,-(SP)        ;;PUSH R1 ON STACK
10046  051214  010246                          MOV     R2,-(SP)        ;;PUSH R2 ON STACK
10047  051216  010346                          MOV     R3,-(SP)        ;;PUSH R3 ON STACK
10048  051220  010446                          MOV     R4,-(SP)        ;;PUSH R4 ON STACK
10049  051222  010546                          MOV     R5,-(SP)        ;;PUSH R5 ON STACK
10050  051224  013700  001652                  MOV     @#RHMR, R0      ;NOW R0 HAS MAINTENANCE REG. ADR.
10051  051230  013703  051202                  MOV     @#SECTR,R3      ;SECTOR COUNTER
10052  051234  012710  000001                  MOV     #DMD,   @R0     ;SET DIAGNOSTIC MODE
10053  051240  052710  000010                  BIS     #MSTCK, @R0     ;SET SECTOR CLOCK
10054  051244  042710  000010                  BIC     #MSTCK, @R0     ;CLEAR SECTOR CLOCK
10055  051250  052710  000010                  BIS     #MSTCK, @R0     ;SET SECTOR CLOCK
10056  051254  042710  000010                  BIC     #MSTCK, @R0     ;CLEAR SECTOR CLOCK
10057                                                                   ;THE ABOVE TWO SECTOR CLOCKS ARE GIVEN FOR
```

```
10058                                                                   ;RESETING SECTOR PULSE
10059                                                                   ;IN CASE IT STARTS SET
10060  051260  052710  000014                BIS     #MINX!MSTCK,@R0 ;SET INDEX AND SECTOR CLOCK
10061  051264  012710  000001                MOV     #DMD,   @R0     ;RESET INDEX AND SECTOR CLOCK
10062  051270  005703                        TST     R3              ;IF SECTOR REQUIRED JUMP OUT
10063  051272  001461                        BEQ     7$              ;BRANCH OF SECTOR ZERO REQUIRED
10064                                        ;NOW THE 304 WORDS WILL START
10065
10066
10067
10068                                ;FOR FIRST BYTE SECTOR CLOCK WILL GO HIGH THEN CLOCK WILL GO HIGH
10069                                ;BOTH WILL COME DOWN TOGETHER THEN SEVEN CLOCKS WILL BE GIVEN
10070                                ;;FOR SECOND BYTE AND ALL OTHER BYTES TILL NEXT SECTOR SECTOR CLOCK
10071                                ;WILL BE IDENTICAL WITH ONE CLOCK
10072
10073
10074                                        ;ONE WORD ONLY
10075
10076  051274  012702  000010        1$:     MOV     #8.,    R2      ;BYTE COUNTER
10077  051300  012705  000002                MOV     #2,     R5      ;BYTES PER WORD
10078  051304  052710  000010                BIS     #MSTCK,@R0      ;SET SECTOR CLOCK
10079  051310  052710  000002                BIS     #MCLK,@R0       ;SET CLOCK
10080  051314  000402                        BR      3$              ;BRANCH TO CLEAR SECTOR AND CLOCK
10081  051316  052710  000012        2$:     BIS     #MSTCK!MCLK,@R0 ;SET SECTOR AND CLOCK
10082  051322  042710  000012        3$:     BIC     #MSTCK!MCLK,@R0 ;CLEAR SECTOR AND CLOCK
10083  051326  052710  000002        8$:     BIS     #MCLK,  @R0     ;SET CLOCK
10084  051332  042710  000002                BIC     #MCLK,  @R0     ;CLEAR CLOCK
10085  051336  005302                        DEC     R2              ;BYTE COUNTER
10086  051340  001372                        BNE     8$              ;BRANCH IF BYTE NOT COMPLETE
10087  051342  012702  000007                MOV     #7,     R2      ;SETUP FOR SECOND BYTE
10088  051346  005305                        DEC     R5              ;IS WORD COMPLETE?
10089  051350  001362                        BNE     2$              ;BRANCH IF NOT COMPLETE
10090                                                                  ;TO GIVE SECTOR CLOCK AND CLOCK
10091
10092
10093                                        ;NOW 303 WORDS ARE LEFT AND ALL ARE IDENTICAL
10094
10095  051352  012701  000457                MOV     #303.,  R1      ;WORDS PER SECTOR COUNTER
10096  051356  012705  000002        4$:     MOV     #2,     R5      ;BYTES PER WORD COUNTER
10097  051362  012702  000007        5$:     MOV     #7,     R2      ;BYTE COUNTER (CLOCK COUNTER)
10098  051366  052710  000012                BIS     #MSTCK!MCLK,@R0 ;SET SECTOR CLOCK AND CLOCK
10099  051372  042710  000012                BIC     #MSTCK!MCLK,@R0 ;CLEAR SECTOR CLOCK AND CLOCK
10100  051376  052710  000002        6$:     BIS     #MCLK,  @R0     ;SET CLOCK
10101  051402  042710  000002                BIC     #MCLK,  @R0     ;RESET CLOCK
10102  051406  005302                        DEC     R2              ;IS BYTE COMPLETE?
10103  051410  001372                        BNE     6$              ;BRANCH IF NOT COMPLETE
10104  051412  005305                        DEC     R5              ;IS WORD COMPLETE?
10105  051414  001362                        BNE     5$              ;BRANCH IF NOT
10106  051416  005301                        DEC     R1              ;IS SECTOR COMPLETE
10107  051420  001356                        BNE     4$              ;BRANCH IF NOT
10108  051422  052710  000010                BIS     #MSTCK,@R0      ;SET SECTOR
10109  051426  042710  000010                BIC     #MSTCK,@R0      ;CLEAR SECTOR
10110  051432  005303                        DEC     R3              ;IS REQUIRED NO OF SECTORS COMPLETE
10111  051434  001317                        BNE     1$              ;BRANCH IF NOT
```

```
10112
10113  051436                          7$:
10114  051436  012605                          MOV     (SP)+,R5        ;;POP STACK INTO R5
10115  051440  012604                          MOV     (SP)+,R4        ;;POP STACK INTO R4
10116  051442  012603                          MOV     (SP)+,R3        ;;POP STACK INTO R3
10117  051444  012602                          MOV     (SP)+,R2        ;;POP STACK INTO R2
10118  051446  012601                          MOV     (SP)+,R1        ;;POP STACK INTO R1
10119  051450  012600                          MOV     (SP)+,R0        ;;POP STACK INTO R0
10120  051452  000201                          RTS     R1
10121
10122
10123                                  ;*****************************************************************
10124                                  ;READ ONE SECTOR OF DATA
10125                                  ;*****************************************************************
10126
10127  051454  000000                  RNO:    0                       ;NO. OF WORDS READ
10128  051456  000000                  RCOM:   0                       ;EXTRA STORAGE
10129
10130
10131
10132  051460  012137  051454          REDATA: MOV     (R1)+,@#RNO     ;SAVE NO. OF WORDS ONLY FOR INFORMATION
10133  051464  012137  051456                  MOV     (R1)+,@#RCOM    ;EXTRA WORD ONLY FOR INFORMATION
10134  051470  010146                          MOV     R1,-(SP)        ;;PUSH R1 ON STACK
10135  051472  005737  002012                  TST     @#TSECC         ;IS THIS AN ECC TEST
10136  051476  001403                          BEQ     1$              ;BRANCH IF NO
10137  051500  012737  177777  043222          MOV     #-1,@#TSECCG    ;THESE BITS ARE TO GENERATE ECC
10138  051506  012702  000402          1$:     MOV     #258.,R2        ;256 WORDS PER SECTOR
10139                                                                  ;PLUS 2 ECC WORDS
10140  051512  012703  047070                  MOV     #DISK,R3        ;POINTE TO DISK SIMULATION
10141  051516  012337  046062          2$:     MOV     (R3)+,@#WORD    ;READY TO READ CONTENTS
10142  051522  004737  046066                  JSR     PC,@#READ       ;READ
10143  051526  005302                          DEC     R2              ;IS 256 WORDS DONE?
10144  051530  001372                          BNE     2$              ;IF NOT BRANCH
10145  051532  005737  002012                  TST     @#TSECC         ;IS THIS AN ECC TEST
10146  051536  001012                          BNE     4$              ;BRANCH OUT IF YES
10147  051540  005037  043222                  CLR     @#TSECCG        ;NO MORE ECC BITS ARE TO BE GENERATED
10148  051544  012702  000017                  MOV     #15.,R2         ;ONE DATA GAP, 14 TOLERANCE GAP
10149  051550  012337  046062          3$:     MOV     (R3)+,@#WORD    ;READY TO READ CONTENTS OF WORD
10150  051554  004737  046066                  JSR     PC,@#READ       ;READ
10151  051560  005302                          DEC     R2              ;COUNT
10152  051562  001372                          BNE     3$              ;BRANCH IF 14 NOT DONE
10153  051564                          4$:
10154  051564  012601                          MOV     (SP)+,R1        ;;POP STACK INTO R1
10155  051566  000201                          RTS     R1              ;RETURN
10156
10157
```

```
10158
10159
10160  051570                          RPVECT:
10161  051570  104400  051576                  TYPE    ,.+4            ;;TYPE ASCIZ STRING
10162  051574  000421                          BR      64$             ;;GET OVER THE ASCIZ
10163                                          ;;.ASCIZ      /UNEXPECTED RP04 INTERRUPT @ PC = /
10164  051640                          64$:
10165  051640  104402                          TYPOC                           ;TYPE FROM PC
10166  051642  012777  051570  127750          MOV     #RPVECT,@RPVEC          ;RESTORE TRAP RP04 VECTOR
10167  051650  000000                          HALT                            ;CHANGE TO CONTINUE
```

```
10168                                   .SBTTL  SYSMAC LIBRARY ROUTINES
10169
10170                                   ;*****************************************************************
10171
10172                                   .SBTTL  SCOPE HANDLER ROUTINE
10173
10174                                   ;*THIS ROUTINE CONTROLS THE LOOPING OF SUBTESTS. IT WILL INCREMENT
10175                                   ;*AND LOAD THE TEST NUMBER($TSTNM) INTO THE DISPLAY REG.(DISPLAY<7:0>)
10176                                   ;*AND LOAD THE ERROR FLAG ($ERFLG) INTO DISPLAY<15:08>
10177                                   ;*THE SWITCH OPTIONS PROVIDED BY THIS ROUTINE ARE:
10178                                   ;*SW14=1        LOOP ON TEST
10179                                   ;*SW11=1        INHIBIT ITERATIONS
10180                                   ;*SW09=1        LOOP ON ERROR
10181                                   ;*SW08=1        LOOP ON TEST IN SWR<7:0>
10182                                   ;*CALL
10183                                   ;*      SCOPE           ;;SCOPE=IOT
10184
10185  051652                           $SCOPE:
10186  051652  005037  045266                 CLR     @#NOSYNC        ;CLEAR FLAG FOR HEADER ERROR COMMANDS
10187  051656  005037  002012                 CLR     @#TSECC         ;CLEAR FLAG FOR ECC TEST
10188                                                                   ;WHEN =177777 IT IS AN ECC TEST
10189                                                                   ;WHEN =0 IT IS NOT AN ECC TEST
10190
10191  051662  005037  043222                 CLR     @#TSECCG        ;EVEN IN AN ECC TEST EVERY CLOCK
10192                                                                   ;IS NOT TO GENERATE ECC
10193                                                                   ;IF =177777 GENERATE ECC
10194                                                                   ;IF =0 DO NOT GENERATE ECC
10195  051666  005037  002014                 CLR     @#TESDTE        ;DRIVE TIMING ERROR TEST
10196  051672                           1$:
10197  051672  006137  177570                 ROL     @#SWR           ;;LOOP ON PRESENT TEST?
10198  051676  100511                         BMI     $OVER           ;;YES IF SW14=1
10199                                   ;#####START OF CODE FOR THE XOR TESTER#####
10200  051700  000416                 $XTSTR: BR      6$              ;;IF RUNNING ON THE "XOR" TESTER CHANGE
10201                                                                   ;;THIS INSTRUCTION TO A "NOP" (NOP=240)
10202  051702  013746  000004                 MOV     @#ERRVEC,-(SP)  ;;SAVE THE CONTENTS OF THE ERROR VECTOR
10203  051706  012737  051726  000004         MOV     #5$,@#ERRVEC    ;;SET FOR TIMEOUT
10204  051714  005737  177060                 TST     @#177060        ;;TIME OUT ON XOR?
10205  051720  012637  000004                 MOV     (SP)+,@#ERRVEC  ;;RESTORE THE ERROR VECTOR
10206  051724  000463                         BR      $SVLAD          ;;GO TO THE NEXT TEST
10207  051726  022626                  5$:    CMP     (SP)+,(SP)+     ;;CLEAR THE STACK AFTER A TIME OUT
10208  051730  012637  000004                 MOV     (SP)+,@#ERRVEC  ;;RESTORE THE ERROR VECTOR
10209  051734  000423                         BR      7$              ;;LOOP ON THE PRESENT TEST
10210  051736                           6$:;#####END OF CODE FOR THE XOR TESTER#####
10211  051736  032737  000400  177570         BIT     #BIT08,@#SWR    ;;LOOP ON SPEC. TEST?
10212  051744  001404                         BEQ     2$              ;;BR IF NO
10213  051746  123767  177570  127126         CMPB    @#SWR,$TSTNM    ;;ON THE RIGHT TEST?    SWR<7:0>
10214  051754  001462                         BEQ     $OVER           ;;BR IF YES
10215  051756  105767  127121          2$:    TSTB    $ERFLG          ;;HAS AN ERROR OCCURRED?
10216  051762  001421                         BEQ     3$              ;;BR IF NO
10217  051764  126767  127125  127111         CMPB    $ERMAX,$ERFLG   ;;MAX. ERRORS FOR THIS TEST OCCURRED?
10218  051772  101015                         BHI     3$              ;;BR IF NO
10219  051774  032737  001000  177570         BIT     #BIT09,@#SWR    ;;LOOP ON ERROR?
10220  052002  001404                         BEQ     4$              ;;BR IF NO
10221  052004  016767  127100  127074  7$:    MOV     $LPERR,$LPADR   ;;SET LOOP ADDRESS TO LAST SCOPE
```

```
10222  052012  000443                          BR      $OVER
10223  052014  105067  127063          4$:     CLRB    $ERFLG          ;;ZERO THE ERROR FLAG
10224  052020  005067  127160                  CLR     $TIMES          ;;CLEAR THE NUMBER OF ITERATIONS TO MAKE
10225  052024  000415                          BR      1$              ;;ESCAPE TO THE NEXT TEST
10226  052026  032737  004000  177570  3$:     BIT     #BIT11,@#SWR    ;;INHIBIT ITERATIONS?
10227  052034  001011                          BNE     1$              ;;BR IF YES
10228  052036  005767  127036                  TST     $PASS           ;;IF FIRST PASS OF PROGRAM
10229  052042  001406                          BEQ     1$              ;;      INHIBIT ITERATIONS
10230  052044  005267  127034                  INC     $ICNT           ;;INCREMENT ITERATION COUNT
10231  052050  026767  127130  127026          CMP     $TIMES,$ICNT    ;;CHECK THE NUMBER OF ITERATIONS MADE
10232  052056  002021                          BGE     $OVER           ;;BR IF MORE ITERATION REQUIRED
10233  052060  012767  000001  127016  1$:     MOV     #1,$ICNT        ;;REINITIALIZE THE ITERATION COUNTER
10234  052066  016767  000044  127110          MOV     $MXCNT,$TIMES   ;;SET NUMBER OF ITERATIONS TO DO
10235  052074  105267  127002          $SVLAD: INCB    $TSTNM          ;;COUNT TEST NUMBERS
10236  052100  011667  127002                  MOV     (SP),$LPADR     ;;SAVE SCOPE LOOP ADDRESS
10237  052104  011667  127000                  MOV     (SP),$LPERR     ;;SAVE ERROR LOOP ADDRESS
10238  052110  005067  127072                  CLR     $ESCAPE         ;;CLEAR THE ESCAPE FROM ERROR ADDRESS
10239  052114  112767  000001  126773          MOVB    #1,$ERMAX       ;;ONLY ALLOW ONE(1) ERROR ON NEXT TEST
10240  052122  016737  126754  177570  $OVER:  MOV     $TSTNM,@#DISPLAY;;DISPLAY TEST NUMBER
10241  052130  016716  126752                  MOV     $LPADR,(SP)     ;;FUDGE RETURN ADDRESS
10242  052134  000002                          RTI                     ;;FIXES PS
10243  052136  000004                  $MXCNT: 4                       ;;MAX. NUMBER OF ITERATIONS
```

```
10244                                   ;*****************************************************************
10245
10246                                   .SBTTL  CONVERT BINARY TO DECIMAL AND TYPE ROUTINE
10247
10248                                   ;*THIS ROUTINE IS USED TO CHANGE A 16-BIT BINARY NUMBER TO A 5-DIGIT
10249                                   ;*SIGNED DECIMAL (ASCII) NUMBER AND TYPE IT. DEPENDING ON WHETHER THE
10250                                   ;*NUMBER IS POSITIVE OR NEGATIVE A SPACE OR A MINUS SIGN WILL BE TYPED
10251                                   ;*BEFORE THE FIRST DIGIT OF THE NUMBER. LEADING ZEROS WILL ALWAYS BE
10252                                   ;*REPLACED WITH SPACES.
10253                                   ;*CALL:
10254                                   ;*      MOV     NUM,-(SP)       ;;PUT THE BINARY NUMBER ON THE STACK
10255                                   ;*      TYPOS                   ;;GO TO THE ROUTINE
10256
10257  052140                           $TYPOS:
10258  052140  010046                          MOV     R0,-(SP)        ;;PUSH R0 ON STACK
10259  052142  010146                          MOV     R1,-(SP)        ;;PUSH R1 ON STACK
10260  052144  010246                          MOV     R2,-(SP)        ;;PUSH R2 ON STACK
10261  052146  010346                          MOV     R3,-(SP)        ;;PUSH R3 ON STACK
10262  052150  010546                          MOV     R5,-(SP)        ;;PUSH R5 ON STACK
10263  052152  012746  020200                  MOV     #20200,-(SP)    ;;SET BLANK SWITCH AND SIGN
10264  052156  016605  000020                  MOV     20(SP),R5       ;;GET THE INPUT NUMBER
10265  052162  100004                          BPL     1$              ;;BR IF INPUT IS POS.
10266  052164  005405                          NEG     R5              ;;MAKE THE BINARY NUMBER POS.
10267  052166  112766  000055  000001          MOVB    #'-,1(SP)       ;;MAKE THE ASCII NUMBER NEG.
10268  052174  005000                  1$:     CLR     R0              ;;ZERO THE CONSTANTS INDEX
10269  052176  012703  052354                  MOV     #$DBLK,R3       ;;SETUP THE OUTPUT POINTER
10270  052202  112723  000040                  MOVB    #' ,(R3)+       ;;SET THE FIRST CHARACTER TO A BLANK
10271  052206  005002                  2$:     CLR     R2              ;;CLEAR THE BCD NUMBER
10272  052210  016001  052344                  MOV     $DTBL(R0),R1    ;;GET THE CONSTANT
10273  052214  160105                  3$:     SUB     R1,R5           ;;FORM THIS BCD DIGIT
10274  052216  002402                          BLT     4$              ;;BR IF DONE
10275  052220  005202                          INC     R2              ;;INCREASE THE BCD DIGIT BY 1
10276  052222  000774                          BR      3$
10277  052224  060105                  4$:     ADD     R1,R5           ;;ADD BACK THE CONSTANT
10278  052226  005702                          TST     R2              ;;CHECK IF BCD DIGIT=0
10279  052230  001002                          BNE     5$              ;;FALL THROUGH IF 0
10280  052232  105716                          TSTB    (SP)            ;;STILL DOING LEADING 0'S?
10281  052234  100407                          BMI     7$              ;;BR IF YES
10282  052236  106316                  5$:     ASLB    (SP)            ;;MSD?
10283  052240  103003                          BCC     6$              ;;BR IF NO
10284  052242  116663  000001  177777          MOVB    1(SP),-1(R3)    ;;YES--SET THE SIGN
10285  052250  052702  000060          6$:     BIS     #'0,R2          ;;MAKE THE BCD DIGIT ASCII
10286  052254  052702  000040          7$:     BIS     #' ,R2          ;;MAKE IT A SPACE IF NOT ALREADY A DIGIT
10287  052260  110223                          MOVB    R2,(R3)+        ;;PUT THIS CHARACTER IN THE OUTPUT BUFFER
10288  052262  005720                          TST     (R0)+           ;;JUST INCREMENTING
10289  052264  020027  000010                  CMP     R0,#10          ;;CHECK THE TABLE INDEX
10290  052270  002746                          BLT     2$              ;;GO DO THE NEXT DIGIT
10291  052272  003002                          BGT     8$              ;;GO TO EXIT
10292  052274  010502                          MOV     R5,R2           ;;GET THE LSD
10293  052276  000764                          BR      6$              ;;GO CHANGE TO ASCII
10294  052300  105726                  8$:     TSTB    (SP)+           ;;WAS THE LSD THE FIRST NON-ZERO?
10295  052302  100003                          BPL     9$              ;;BR IF NO
10296  052304  116663  177777  177776          MOVB    -1(SP),-2(R3)   ;;YES--SET THE SIGN FOR TYPING
10297  052312  105013                  9$:     CLRB    (R3)            ;;SET THE TERMINATOR
```

```
10298  052314  012605                          MOV     (SP)+,R5        ;;POP STACK INTO R5
10299  052316  012603                          MOV     (SP)+,R3        ;;POP STACK INTO R3
10300  052320  012602                          MOV     (SP)+,R2        ;;POP STACK INTO R2
10301  052322  012601                          MOV     (SP)+,R1        ;;POP STACK INTO R1
10302  052324  012600                          MOV     (SP)+,R0        ;;POP STACK INTO R0
10303  052326  104400  052354                  TYPE    ,$DBLK          ;;NOW TYPE THE NUMBER
10304  052332  016666  000002  000004          MOV     2(SP),4(SP)     ;;ADJUST THE STACK
10305  052340  012616                          MOV     (SP)+,(SP)
10306  052342  000002                          RTI                     ;;RETURN TO USER
10307  052344  023420                $DTBL:    10000.
10308  052346  001750                          1000.
10309  052350  000144                          100.
10310  052352  000012                          10.
10311  052354  000004                $DBLK:    .BLKW   4
```

```
10312                                   ;*****************************************************************
10313
10314                                   .SBTTL  TYPE ROUTINE
10315
10316                                   ;*ROUTINE TO TYPE ASCIZ MESSAGE. MESSAGE MUST TERMINATE WITH A 0 BYTE.
10317                                   ;*THE ROUTINE WILL INSERT A NUMBER OF NULL CHARACTERS AFTER A LINE FEED.
10318                                   ;*NOTE1:        $NULL CONTAINS THE CHARACTER TO BE USED AS THE FILLER CHARACTER.
10319                                   ;*NOTE2:        $FILLS CONTAINS THE NUMBER OF FILLER CHARACTERS REQUIRED.
10320                                   ;*NOTE3:        $FILLC CONTAINS THE CHARACTER TO FILL AFTER.
10321                                   ;*
10322                                   ;*CALL:
10323                                   ;*1) USING A TRAP INSTRUCTION
10324                                   ;*      TYPE    ,MESADR         ;;MESADR IS FIRST ADDRESS OF AN ASCIZ STRING
10325                                   ;*OR
10326                                   ;*      TYPE
10327                                   ;*      MESADR
10328                                   ;*
10329                                   ;*2) USING A JSR INSTRUCTION
10330                                   ;*      MOV     PS,-(SP)        ;;PUSH PROCESSOR STATUS WORD ON THE STACK
10331                                   ;*      JSR     PC,$TYPE        ;;CALL TYPE ROUTINE
10332                                   ;*      MESADDR                 ;;FIRST ADRESS OF MESSAGE
10333
10334  052364  105767  126561          $TYPE:  TSTB    $TPFLG          ;;IS THERE A TERMINAL?
10335  052370  100002                          BPL     1$              ;;BR IF YES
10336  052372  000000                          HALT                    ;;HALT HERE IF NO TERMINAL
10337  052374  000407                          BR      3$              ;;LEAVE
10338  052376  010046                  1$:     MOV     R0,-(SP)        ;;SAVE R0
10339  052400  017600  000002                  MOV     @2(SP),R0       ;;GET ADDRESS OF ASCIZ STRING
10340  052404  112046                  2$:     MOVB    (R0)+,-(SP)     ;;PUSH CHARACTER TO BE TYPED ONTO STACK
10341  052406  001005                          BNE     4$              ;;BR IF IT ISN'T THE TERMINATOR
10342  052410  005726                          TST     (SP)+           ;;IF TERMINATOR POP IT OFF THE STACK
10343  052412  012600                          MOV     (SP)+,R0        ;;RESTORE R0
10344  052414  062716  000002          3$:     ADD     #2,(SP)         ;;ADJUST RETURN PC
10345  052420  000002                          RTI                     ;;RETURN
10346  052422  004767  000026          4$:     JSR     PC,$TYPEC       ;;GO TYPE THIS CHARACTER
10347  052426  126726  126516          5$:     CMPB    $FILLC,(SP)+    ;;IS IT TIME FOR FILLER CHARS.?
10348  052432  001364                          BNE     2$              ;;IF NO GO GET NEXT CHAR.
10349  052434  016746  126506                  MOV     $NULL,-(SP)     ;;GET # OF FILLER CHARS. NEEDED
10350                                                                  ;;AND THE NULL CHAR.
10351  052440  105366  000001          6$:     DECB    1(SP)           ;;DOES A NULL NEED TO BE TYPED?
10352  052444  002770                          BLT     5$              ;;BR IF NO--GO POP THE NULL OFF OF STACK
10353  052446  004767  000002                  JSR     PC,$TYPEC       ;;GO TYPE A NULL
10354  052452  000772                          BR      6$              ;;LOOP
10355  052454  105777  126462          $TYPEC: TSTB    @$TPS           ;;WAIT UNTIL PRINTER IS READY
10356  052460  100375                          BPL     $TYPEC
10357  052462  116677  000002  126454        MOVB    2(SP),@$TPB     ;;LOAD CHAR TO BE TYPED INTO DATA REG.
10358  052470  000207                          RTS     PC
```

```
10359                                   ;*****************************************************************
10360
10361                                   .SBTTL  TTY INPUT ROUTINE
10362
10363  052472  000000                   $TKCNT: .WORD   0               ;;NUMBER OF ITEMS IN QUEUE
10364  052474  000000                   $TKQIN: .WORD   0               ;;INPUT POINTER
10365  052476  000000                   $TKQOUT: .WORD  0               ;;OUTPUT POINTER
10366  052500  000011                   $TKQSRT: .BLKB  9.              ;;TTY KEYBOARD QUEUE
10367          052511                   $TKQEND=.
10368          052512                   .EVEN
10369
10370                                   ;*TK INITIALIZE ROUTINE
10371                                   ;*THIS ROUTINE WILL INITIALIZE THE TTY KEYBOARD INPUT QUEUE
10372                                   ;*SETUP THE INTERRUPT VECTOR AND TURN ON THE KEYBOARD INTERRUPT
10373                                   ;
10374                                   ;*CALL
10375                                   ;*      JSR     PC,$TKINT
10376                                   ;*      RETURN
10377                                   ;
10378  052512  005067  177754           $TKINT: CLR     $TKCNT          ;;CLEAR COUNT OF ITEMS IN QUEUE
10379  052516  012767  052500  177750           MOV     #$TKQSRT,$TKQIN ;;MOVE THE STARTING ADDRESS OF THE
10380  052524  016767  177744  177744           MOV     $TKQIN,$TKQOUT  ;;QUEUE INTO THE INPUT & OUTPUT POINTERS.
10381  052532  012737  052562  000060           MOV     #$TKSRV,@#TKVEC ;;INITIALIZE THE KEYBOARD VECTOR
10382  052540  012737  000200  000062           MOV     #200,@#TKVEC+2  ;;"BR" LEVEL 4
10383  052546  005777  126366                   TST     @$TKB           ;;CLEAR DONE FLAG
10384  052552  012777  000100  126356           MOV     #BIT06,@$TKS    ;;ENABLE INTERRUPT
10385  052560  000207                           RTS     PC              ;;RETURN TO CALLER
10386
10387                                   ;*TK SERVICE ROUTINE
10388                                   ;*THIS ROUTINE WILL SERVICE THE TTY KEYBOARD INTERRUPT
10389                                   ;
10390  052562  117746  126352           $TKSRV: MOVB    @$TKB,-(SP)     ;;PICKUP THE CHARACTER
10391  052566  042716  177600                   BIC     #^C177,(SP)     ;;STRIP THE JUNK
10392  052572  021627  000003                   CMP     (SP),#3         ;;IS IT A CONTROL C?
10393  052576  001006                           BNE     1$              ;;BRANCH IF NO
10394  052600  104400  053073                   TYPE    ,$CNTLC         ;;TYPE A CONTROL-C (^C)
10395  052604  004767  177702                   JSR     PC,$TKINT       ;;INIT THE KEYBOARD
10396  052610  000167  164532                   JMP     OPERSEL         ;;CONTROL C RESTART
10397  052614  022767  000011  177650   1$:     CMP     #9.,$TKCNT      ;;IS THE QUEUE FULL?
10398  052622  001004                           BNE     2$              ;;BRANCH IF NO
10399  052624  104400  001210                   TYPE    ,$BELL          ;;RING THE TTY BELL
10400  052630  005726                           TST     (SP)+           ;;CLEAN CHARACTER OFF OF STACK
10401  052632  000002                           RTI                     ;;RETURN
10402  052634  005267  177632           2$:     INC     $TKCNT          ;;COUNT THIS CHARACTER
10403  052640  112677  177630                   MOVB    (SP)+,@$TKQIN   ;;AND PUT IT IN QUEUE
10404  052644  005267  177624                   INC     $TKQIN          ;;UPDATE THE POINTER
10405  052650  026727  177620  052511           CMP     $TKQIN,#$TKQEND ;;GO OFF THE END?
10406  052656  001003                           BNE     3$              ;;BRANCH IF NO
10407  052660  012767  052500  177606           MOV     #$TKQSRT,$TKQIN ;;RESET THE POINTER
10408  052666  000002                   3$:     RTI                     ;;RETURN
10409                                   ;*****************************************************************
10410                                   ;*THIS ROUTINE WILL INPUT A SINGLE CHARACTER FROM THE TTY
10411                                   ;*CALL:
10412                                   ;*      RDCHR                   ;;INPUT A SINGLE CHARACTER FROM THE TTY
```

```
10413                                   ;*      RETURN HERE             ;;CHARACTER IS ON THE STACK
10414                                   ;
10415
10416  052670  011646                   $RDCHR: MOV     (SP),-(SP)      ;;PUSH DOWN THE PC AND
10417  052672  016666  000004  000002           MOV     4(SP),2(SP)     ;;THE PS
10418  052700  005046  000004                   CLR     4(SP)           ;;GET READY FOR A CHARACTER
10419  052704  005C37  177776                   CLR     @#PS            ;;ALLOW INTERRUPTS
10420  052710  005767  177556           1$:     TST     $TKCNT          ;;WAIT ON A CHARACTER
10421  052714  001775                   BEQ     1$
10422  052716  005367  177550                   DEC     $TKCNT          ;;DECREMENT THE COUNTER
10423  052722  117766  177550  000004           MOVB    @$TKQOUT,4(SP)  ;;GET ONE CHARACTER
10424  052730  005267  177542                   INC     $TKQOUT         ;;UPDATE THE POINTER
10425  052734  026727  177536  052511           CMP     $TKQOUT,#$TKQEND ;;DID IT GO OFF OF THE END?
10426  052742  001003                   BNE     2$              ;;BRANCH IF NO
10427  052744  012767  052500  177524           MOV     #$TKQSRT,$TKQOUT ;;RESET THE POINTER
10428  052752  000002                   2$:     RTI                     ;;RETURN
10429                                   ;*****************************************************************
10430                                   ;*THIS ROUTINE WILL INPUT A STRING FROM THE TTY
10431                                   ;*CALL:
10432                                   ;*      RDLIN                   ;;INPUT A STRING FROM THE TTY
10433                                   ;*      RETURN HERE             ;;ADDRESS OF FIRST CHARACTER WILL BE ON THE STACK
10434                                   ;*                              ;;TERMINATOR WILL BE A BYTE OF ALL 0'S
10435
10436  052754  010346                   $RDLIN: MOV     R3,-(SP)        ;;SAVE R3
10437  052756  012703  053062           1$:     MOV     #$TTYIN,R3      ;;GET ADDRESS
10438  052762  022703  053073           2$:     CMP     #$TTYIN+9.,R3   ;;BUFFER FULL?
10439  052766  101405                   BLOS    4$              ;;BR IF YES
10440  052770  104412                   RDCHR                   ;;GO READ ONE CHARACTER FROM THE TTY
10441  052772  112613                   MOVB    (SP)+,(R3)      ;;GET CHARACTER
10442  052774  122713  000177           CMPB    #177,(R3)       ;;IS IT A RUBOUT
10443  053000  001003                   BNE     3$              ;;SKIP IF NOT
10444  053002  104400  001214           4$:     TYPE    $QUES           ;;TYPE A '?'
10445  053006  000763                   BR      1$              ;;CLEAR THE BUFFER AND LOOP
10446  053010  111367  000044           3$:     MOVB    (R3),9$         ;;ECHO THE CHARACTER
10447  053014  104400  053060                   TYPE    9$
10448  053020  122723  000015                   CMPB    #15,(R3)+       ;;CHECK FOR RETURN
10449  053024  001356                   BNE     2$              ;;LOOP IF NOT RETURN
10450  053026  105063  177777                   CLRB    -1(R3)          ;;CLEAR RETURN (THE 15)
10451  053032  104400  001216                   TYPE    $LF             ;;TYPE A LINE FEED
10452  053036  012603                   MOV     (SP)+,R3        ;;RESTORE R3
10453  053040  011646                   MOV     (SP),-(SP)      ;;ADJUST THE STACK AND PUT ADDRESS OF THE
10454  053042  016666  000004  000002           MOV     4(SP),2(SP)     ;;      FIRST ASCII CHARACTER ON IT
10455  053050  012766  053062  000004           MOV     #$TTYIN,4(SP)
10456  053056  000002                   RTI                     ;;RETURN
10457  053060     000                   9$:     .BYTE   0               ;;STORAGE FOR ASCII CHAR. TO TYPE
10458  053061     000                   .BYTE   0               ;;TERMINATOR
10459  053062  000011                   $TTYIN: .BLKB   9.              ;;RESERVE 9. BYTES FOR TTY INPUT
10460  053073     136  006503  000012  $CNTLC: .ASCIZ  /^C/<15><12>    ;;CONTROL "C"
10461                                                                   ;FROM THE TTY
```

```
10462                                   ;*****************************************************************
10463
10464                                   .SBTTL  READ AN OCTAL NUMBER FROM THE TTY
10465
10466                                   ;*THIS ROUTINE WILL READ AN OCTAL (ASCII) NUMBER FROM THE TTY AND
10467                                   ;*CHANGE IT TO BINARY.
10468                                   ;*THE INPUT CHARACTERS WILL BE CHECKED TO INSURED THEY ARE LEGAL
10469                                   ;*OCTAL DIGITS. IF AN ILLEGAL CHARACTER IS READ A "?" WILL BE TYPED
10470                                   ;*FOLLOWED BY A CARRIAGE RETURN-LINE FEED. THE COMPLETE NUMBER MUST
10471                                   ;*THEN BE RETYPED. THE INPUT IS TERMINATED BY TYPING A CARRIAGE RETURN.
10472                                   ;*CALL:
10473                                   ;*      RDOCT                   ;;READ AN OCTAL NUMBER
10474                                   ;*      RETURN HERE             ;;LOW ORDER BITS ARE ON TOP OF THE STACK
10475                                   ;*                              ;;HIGH ORDER BITS ARE IN $HIOCT
10476
10477  053100  011646                  $RDOCT: MOV     (SP),-(SP)      ;;PROVIDE SPACE FOR THE
10478  053102  016666  000004  000002          MOV     4(SP),2(SP)     ;;INPUT NUMBER
10479  053110  010046                          MOV     R0,-(SP)        ;;PUSH R0 ON STACK
10480  053112  010146                          MOV     R1,-(SP)        ;;PUSH R1 ON STACK
10481  053114  010246                          MOV     R2,-(SP)        ;;PUSH R2 ON STACK
10482  053116  104414                  1$:     RDLIN                   ;;READ AN ASCIZ LINE
10483  053120  012600                          MOV     (SP)+,R0        ;;GET ADDRESS OF 1ST CHARACTER
10484  053122  010067  000100                  MOV     R0,5$           ;;AND SAVE IT
10485  053126  005001                          CLR     R1              ;;CLEAR DATA WORD
10486  053130  005002                          CLR     R2
10487  053132  112046                  2$:     MOVB    (R0)+,-(SP)     ;;PICKUP THIS CHARACTER
10488  053134  001420                          BEQ     3$              ;;IF ZERO GET OUT
10489  053136  122716  000060                  CMPB    #'0,(SP)        ;;MAKE SURE THIS CHARACTER
10490  053142  003026                          BGT     4$              ;;IS AN OCTAL DIGIT
10491  053144  122716  000067                  CMPB    #'7,(SP)
10492  053150  002423                          BLT     4$
10493  053152  006301                          ASL     R1              ;;*2
10494  053154  006102                          ROL     R2
10495  053156  006301                          ASL     R1              ;;*4
10496  053160  006102                          ROL     R2
10497  053162  006301                          ASL     R1              ;;*8
10498  053164  006102                          ROL     R2
10499  053166  042716  177770                  BIC     #^C7,(SP)       ;;STRIP THE ASCII JUNK
10500  053172  062601                          ADD     (SP)+,R1        ;;ADD IN THIS DIGIT
10501  053174  000756                          BR      2$              ;;LOOP
10502  053176  005726                  3$:     TST     (SP)+           ;;CLEAN TERMINATOR FROM STACK
10503  053200  010166  000012                  MOV     R1,12(SP)       ;;SAVE THE RESULT
10504  053204  010267  000026                  MOV     R2,$HIOCT
10505  053210  012602                          MOV     (SP)+,R2        ;;POP STACK INTO R2
10506  053212  012601                          MOV     (SP)+,R1        ;;POP STACK INTO R1
10507  053214  012600                          MOV     (SP)+,R0        ;;POP STACK INTO R0
10508  053216  000002                          RTI                     ;;RETURN
10509  053220  005726                  4$:     TST     (SP)+           ;;CLEAN PARTIAL FROM STACK
10510  053222  105010                          CLRB    (R0)            ;;SET A TERMINATOR
10511  053224  104400                          TYPE                    ;;TYPE UP THRU THE BAD CHAR.
10512  053226  000000                  5$:     .WORD   0
10513  053230  104400  001214                  TYPE    $QUES           ;;"?" "CR" & "LF"
10514  053234  000730                          BR      1$              ;;TRY AGAIN
10515  053236  000000                  $HIOCT: .WORD   0               ;;HIGH ORDER BITS GO HERE
```

```
10516                                   ;*****************************************************************
10517
10518                                   .SBTTL  ERROR HANDLER ROUTINE
10519
10520                                   ;*THIS ROUTINE WILL INCREMENT THE ERROR FLAG AND THE ERROR COUNT,
10521                                   ;*SAVE THE ERROR ITEM NUMBER AND THE ADDRESS OF THE ERROR CALL
10522                                   ;*AND GO TO $ERRTYP ON ERROR
10523                                   ;*THE SWITCH OPTIONS PROVIDED BY THIS ROUTINE ARE:
10524                                   ;*SW15=1        HALT ON ERROR
10525                                   ;*SW13=1        INHIBIT ERROR TYPEOUTS
10526                                   ;*SW10=1        BELL ON ERROR
10527                                   ;*SW09=1        LOOP ON ERROR
10528                                   ;*CALL
10529                                   ;*      ERROR   N       ;;ERROR=EMT AND N=ERROR ITEM NUMBER
10530
10531   053240                          $ERROR:
10532   053240  012737  177777  001774          MOV     #-1,@#ERFLGS    ;SET ERROR FLAG
10533   053246                          REGSA1:
10534   053246  105267  125631          7$:     INCB    $ERFLG          ;;SET THE ERROR FLAG
10535   053252  001775                          BEQ     7$              ;;DON'T LET THE FLAG GO TO ZERO
10536   053254  016737  125622  177570          MOV     $TSTNM,@#DISPLAY;;DISPLAY TEST NUMBER AND ERROR FLAG
10537   053262  032737  002000  177570          BIT     #BIT10,@#SWR    ;;BELL ON ERROR?
10538   053270  001402                          BEQ     1$              ;;NO - SKIP
10539   053272  104400  001210                  TYPE    ,$BELL          ;;RING BELL
10540   053276  005267  125610          1$:     INC     $ERTTL          ;;COUNT THE NUMBER OF ERRORS
10541   053302  011667  125610                  MOV     (SP),$ERRPC     ;;GET ADDRESS OF ERROR INSTRUCTION
10542   053306  162767  000002  125602          SUB     #2,$ERRPC
10543   053314  117767  125576  125572          MOVB    @$ERRPC,$ITEMB  ;;STRIP AND SAVE THE ERROR ITEM CODE
10544   053322  032737  020000  177570          BIT     #BIT13,@#SWR    ;;SKIP TYPEOUT IF SET
10545   053330  001004                          BNE     2$              ;;SKIP TYPEOUTS
10546   053332  004737  053402                  JSR     PC,@#$ERRTYP    ;;GO TO USER ERROR ROUTINE
10547   053336  104400  001215                  TYPE    ,$CRLF
10548   053342  005737  177570          2$:     TST     @#SWR           ;;HALT ON ERROR
10549   053346  100001                          BPL     3$              ;;SKIP IF CONTINUE
10550   053350  000000                          HALT                    ;;HALT ON ERROR!
10551   053352  032737  001000  177570  3$:     BIT     #BIT09,@#SWR    ;;LOOP ON ERROR SWITCH SET?
10552   053360  001402                          BEQ     4$              ;;BR IF NO
10553   053362  016716  125522                  MOV     $LPERR,(SP)     ;;FUDGE RETURN FOR LOOPING
10554   053366  005767  125614          4$:     TST     $ESCAPE         ;;CHECK FOR AN ESCAPE ADDRESS
10555   053372  001402                          BEQ     5$              ;;BR IF NONE
10556   053374  016716  125606                  MOV     $ESCAPE,(SP)    ;;FUDGE RETURN ADDRESS FOR ESCAPE
10557   053400                          5$:
10558   053400  000002                          RTI                     ;;RETURN
```

```
10559                                   ;*****************************************************************
10560
10561                                   .SBTTL  ERROR MESSAGE TYPEOUT ROUTINE
10562
10563                                   ;*THIS ROUTINE USES THE "ITEM CONTROL BYTE" ($ITEMB) TO DETERMINE WHICH
10564                                   ;*ERROR IS TO BE REPORTED. IT THEN OBTAINS, FROM THE "ERROR TABLE" ($ERRTB),
10565                                   ;*AND REPORTS THE APPROPRIATE INFORMATION CONCERNING THE ERROR.
10566
10567   053402                          $ERRTYP:
10568   053402  104400  001215                  TYPE    ,$CRLF          ;;"CARRIAGE RETURN" & "LINE FEED"
10569   053406  010046                  MOV     R0,-(SP)        ;;SAVE R0
10570   053410  005000                  CLR     R0              ;;PICKUP THE ITEM INDEX
10571   053412  153700  001114          BISB    @#$ITEMB,R0
10572   053416  001004                  BNE     1$              ;;IF ITEM NUMBER IS ZERO, JUST
10573                                                           ;;TYPE THE PC OF THE ERROR
10574   053420  016746  125472          MOV     $ERRPC,-(SP)    ;;SAVE $ERRPC FOR TYPEOUT
10575                                                           ;;ERROR ADDRESS
10576   053424  104402                  TYPOC                   ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
10577   053426  000445                  BR      10$             ;;GET OUT
10578   053430  005300          1$:     DEC     R0              ;;ADJUST THE INDEX SO THAT IT WILL
10579   053432  006300                  ASL     R0              ;;      WORK FOR THE ERROR TABLE
10580   053434  006300                  ASL     R0
10581   053436  006300                  ASL     R0
10582   053440  062700  001220          ADD     #$ERRTB,R0      ;;FORM TABLE POINTER
10583   053444  012067  000004          MOV     (R0)+,2$        ;;PICKUP "ERROR MESSAGE" POINTER
10584   053450  001404                  BEQ     3$              ;;SKIP TYPEOUT IF NO POINTER
10585   053452  104400                  TYPE                    ;;TYPE THE "ERROR MESSAGE"
10586   053454  000000          2$:     .WORD   0               ;;"ERROR MESSAGE" POINTER GOES HERE
10587   053456  104400  001215          TYPE    ,$CRLF          ;;"CARRIAGE RETURN" & "LINE FEED"
10588   053462  012067  000004  3$:     MOV     (R0)+,4$        ;;PICKUP "DATA HEADER" POINTER
10589   053466  001404                  BEQ     5$              ;;SKIP TYPEOUT IF 0
10590   053470  104400                  TYPE                    ;;TYPE THE "DATA HEADER"
10591   053472  000000          4$:     .WORD   0               ;;"DATA HEADER" POINTER GOES HERE
10592   053474  104400  001215          TYPE    ,$CRLF          ;;"CARRIAGE RETURN" & "LINE FEED"
10593   053500  010146          5$:     MOV     R1,-(SP)        ;;SAVE R1
10594   053502  012001                  MOV     (R0)+,R1        ;;PICKUP "DATA TABLE" POINTER
10595   053504  001415                  BEQ     9$              ;;BR IF NO DATA TO BE TYPED
10596   053506  012000                  MOV     (R0)+,R0        ;;PICKUP "DATA FORMAT" POINTER
10597   053510  105720          6$:     TSTB    (R0)+           ;;"OCTAL" OR "DECIMAL"
10598   053512  001003                  BNE     7$              ;;BR IF DECIMAL
10599   053514  013146                  MOV     @(R1)+,-(SP)    ;;SAVE @(R1)+ FOR TYPEOUT
10600   053516  104402                  TYPOC                   ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
10601   053520  000402                  BR      8$
10602   053522                  7$:
10603   053522  013146                  MOV     @(R1)+,-(SP)    ;;SAVE @(R1)+ FOR TYPEOUT
10604   053524  104410                  TYPDS                   ;;GO TYPE--DECIMAL ASCII WITH SIGN
10605   053526  005711          8$:     TST     (R1)            ;;IS THERE ANOTHER NUMBER?
10606   053530  001403                  BEQ     9$              ;;BR IF NO
10607   053532  104400  053552          TYPE    ,11$            ;;TYPE TWO(2) SPACES
10608   053536  000764                  BR      6$              ;;LOOP
10609
10610   053540  012601          9$:     MOV     (SP)+,R1        ;;RESTORE R1
10611   053542  012600          10$:    MOV     (SP)+,R0        ;;RESTORE R0
10612   053544  104400  001215          TYPE    ,$CRLF          ;;"CARRIAGE RETURN" & "LINE FEED"
```

```
10613  053550  000207                          RTS     PC              ;;RETURN
10614  053552  020040      000         11$:    .ASCIZ  / /             ;;TWO(2) SPACES
10615          053556                          .EVEN
```

```
10616                                   ;*****************************************************************
10617
10618                                   .SBTTL  BINARY TO OCTAL (ASCII) AND TYPE
10619
10620                                   ;*THIS ROUTINE IS USED TO CHANGE A 16-BIT BINARY NUMBER TO A 6-DIGIT
10621                                   ;*OCTAL (ASCII) NUMBER AND TYPE IT.
10622                                   ;*$TYPOS---ENTER HERE TO SETUP SUPPRESS ZEROS AND NUMBER OF DIGITS TO TYPE
10623                                   ;*CALL:
10624                                   ;*      MOV     NUM,-(SP)       ;;NUMBER TO BE TYPED
10625                                   ;*      TYPOS                   ;;CALL FOR TYPEOUT
10626                                   ;*      .BYTE   N               ;;N=1 TO 6 FOR NUMBER OF DIGITS TO TYPE
10627                                   ;*      .BYTE   M               ;;M=1 OR 0
10628                                   ;*                                      ;;1=TYPE LEADING ZEROS
10629                                   ;*                                      ;;0=SUPPRESS LEADING ZEROS
10630                                   ;*
10631                                   ;*$TYPON----ENTER HERE TO TYPE OUT WITH THE SAME PARAMETERS AS THE LAST
10632                                   ;*$TYPOS OR $TYPOC
10633                                   ;*CALL:
10634                                   ;*      MOV     NUM,-(SP)       ;;NUMBER TO BE TYPED
10635                                   ;*      TYPON                   ;;CALL FOR TYPEOUT
10636                                   ;*
10637                                   ;*$TYPOC---ENTER HERE FOR TYPEOUT OF A 16 BIT NUMBER
10638                                   ;*CALL:
10639                                   ;*      MOV     NUM,-(SP)       ;;NUMBER TO BE TYPED
10640                                   ;*      TYPOC                   ;;CALL FOR TYPEOUT
10641
10642  053556  017646  000000       $TYPOS: MOV     @(SP),-(SP)     ;;PICKUP THE MODE
10643  053562  116667  000001  000211       MOVB    1(SP),$OFILL    ;;LOAD ZERO FILL SWITCH
10644  053570  112667  000207               MOVB    (SP)+,$OMODE+1  ;;NUMBER OF DIGITS TO TYPE
10645  053574  062716  000002               ADD     #2,(SP)         ;;ADJUST RETURN ADDRESS
10646  053600  000406                       BR      $TYPON
10647  053602  112767  000001  000171 $TYPOC: MOVB    #1,$OFILL       ;;SET THE ZERO FILL SWITCH
10648  053610  112767  000006  000165       MOVB    #6,$OMODE+1     ;;SET FOR SIX(6) DIGITS
10649  053616  112767  000005  000154 $TYPON: MOVB    #5,$OCNT        ;;SET THE ITERATION COUNT
10650  053624  010346                       MOV     R3,-(SP)        ;;SAVE R3
10651  053626  010446                       MOV     R4,-(SP)        ;;SAVE R4
10652  053630  010546                       MOV     R5,-(SP)        ;;SAVE R5
10653  053632  116704  000145               MOVB    $OMODE+1,R4     ;;GET THE NUMBER OF DIGITS TO TYPE
10654  053636  005404                       NEG     R4
10655  053640  062704  000006               ADD     #6,R4           ;;SUBTRACT IT FOR MAX. ALLOWED
10656  053644  110467  000132               MOVB    R4,$OMODE       ;;SAVE IT FOR USE
10657  053650  116704  000125               MOVB    $OFILL,R4       ;;GET THE ZERO FILL SWITCH
10658  053654  016605  000012               MOV     12(SP),R5       ;;PICKUP THE INPUT NUMBER
10659  053660  005003                       CLR     R3              ;;CLEAR THE OUTPUT WORD
10660  053662  006105              1$:      ROL     R5              ;;ROTATE MSB INTO "C"
10661  053664  000404                       BR      3$              ;;GO DO MSB
10662  053666  006105              2$:      ROL     R5              ;;FORM THIS DIGIT
10663  053670  006105                       ROL     R5
10664  053672  006105                       ROL     R5
10665  053674  010503                       MOV     R5,R3
10666  053676  006103              3$:      ROL     R3              ;;GET LSB OF THIS DIGIT
10667  053700  105367  000076               DECB    $OMODE          ;;TYPE THIS DIGIT?
10668  053704  100016                       BPL     7$              ;;BR IF NO
10669  053706  042703  177770               BIC     #177770,R3      ;;GET RID OF JUNK
```

```
10670  053712  001002                          BNE     4$              ;;TEST FOR 0
10671  053714  005704                          TST     R4              ;;SUPPRESS THIS 0?
10672  053716  001403                          BEQ     5$              ;;BR IF YES
10673  053720  005204                  4$:     INC     R4              ;;DON'T SUPPRESS ANYMORE 0'S
10674  053722  052703  000060                  BIS     #'0,R3          ;;MAKE THIS DIGIT ASCII
10675  053726  052703  000040          5$:     BIS     #' ,R3          ;;MAKE ASCII IF NOT ALREADY
10676  053732  110367  000040                  MOVB    R3,B$           ;;SAVE FOR TYPING
10677  053736  104400  053776                  TYPE    ,B$             ;;GO TYPE THIS DIGIT
10678  053742  105367  000032          7$:     DECB    $OCNT           ;;COUNT BY 1
10679  053746  003347                          BGT     2$              ;;BR IF MORE TO DO
10680  053750  002402                          BLT     6$              ;;BR IF DONE
10681  053752  005204                          INC     R4              ;;INSURE LAST DIGIT ISN'T A BLANK
10682  053754  000744                          BR      2$              ;;GO DO THE LAST DIGIT
10683  053756  012605                  6$:     MOV     (SP)+,R5        ;;RESTORE R5
10684  053760  012604                          MOV     (SP)+,R4        ;;RESTORE R4
10685  053762  012603                          MOV     (SP)+,R3        ;;RESTORE R3
10686  053764  016666  000002  000004          MOV     2(SP),4(SP)     ;;SET THE STACK FOR RETURNING
10687  053772  012616                          MOV     (SP)+,(SP)
10688  053774  000002                          RTI                     ;;RETURN
10689  053776     000          B$:     .BYTE   0               ;;STORAGE FOR ASCII DIGIT
10690  053777     000                  .BYTE   0               ;;TERMINATOR FOR TYPE ROUTINE
10691  054000     000          $OCNT:  .BYTE   0               ;;OCTAL DIGIT COUNTER
10692  054001     000          $OFILL: .BYTE   0               ;;ZERO FILL SWITCH
10693  054002  000000          $OMODE: .WORD   0               ;;NUMBER OF DIGITS TO TYPE
```

```
10694                                   ;*****************************************************************
10695
10696                                   .SBTTL  TRAP DECODER
10697
10698                                   ;*THIS ROUTINE WILL PICKUP THE LOWER BYTE OF THE "TRAP" INSTRUCTION
10699                                   ;*AND USE IT TO INDEX THROUGH THE TRAP TABLE FOR THE STARTING ADDRESS
10700                                   ;*OF THE DESIRED ROUTINE. THEN USING THE ADDRESS OBTAINED IT WILL
10701                                   ;*GO TO THAT ROUTINE.
10702
10703  054004  010046                   $TRAP:  MOV     R0,-(SP)        ;;SAVE R0
10704  054006  016600  000002                   MOV     2(SP),R0        ;;GET TRAP ADDRESS
10705  054012  005740                           TST     -(R0)           ;;BACKUP BY 2
10706  054014  111000                           MOVB    (R0),R0         ;;GET RIGHT BYTE OF TRAP
10707  054016  016000  054024                   MOV     $TRPAD(R0),R0   ;;INDEX TO TABLE
10708  054022  000200                           RTS     R0              ;;GO TO ROUTINE
10709
10710
10711                                   .SBTTL  TRAP TABLE
10712
10713                                   ;*THIS TABLE CONTAINS THE STARTING ADDRESSES OF THE ROUTINES CALLED
10714                                   ;*BY THE "TRAP" INSTRUCTION.
10715
10716                                   ;       ROUTINE
10717                                   ;       -------
10718  054024                           $TRPAD:
10719  054024  052364                           $TYPE   ;;CALL=TYPE     TRAP+0(104400)  TTY TYPEOUT ROUTINE
10720  054026  053602                           $TYPOC  ;;CALL=TYPOC    TRAP+2(104402)  TYPE OCTAL NUMBER (WITH LEADING ZEROS)
10721  054030  053556                           $TYPOS  ;;CALL=TYPOS    TRAP+4(104404)  TYPE OCTAL NUMBER (NO LEADING ZEROS)
10722  054032  053616                           $TYPON  ;;CALL=TYPON    TRAP+6(104406)  TYPE OCTAL NUMBER (AS PER LAST CALL)
10723  054034  052140                           $TYPDS  ;;CALL=TYPDS    TRAP+10(104410) TYPE DECIMAL NUMBER (WITH SIGN)
10724  054036  052670                           $RDCHR  ;;CALL=RDCHR    TRAP+12(104412) TTY TYPEIN CHARACTER ROUTINE
10725  054040  052754                           $RDLIN  ;;CALL=RDLIN    TRAP+14(104414) TTY TYPEIN STRING ROUTINE
10726  054042  053100                           $RDOCT  ;;CALL=RDOCT    TRAP+16(104416) READ AN OCTAL NUMBER FROM TTY
10727  054044  040504                           T.SCOP  ;;CALL=SCOP1    TRAP+20(104420) MY LOCAL SCOPES
10728  054046  040556                           CHECKT  ;;CALL=CHECKD   TRAP+22(104422) CHECK DVA,RDY,DPR,DRY
10729  054050  041074                           WAIT.T  ;;CALL=WAT      TRAP+24(104424) WAIT LOOP
```

```
10730                                   ;*****************************************************************
10731
10732                                   .SBTTL  POWER DOWN AND UP ROUTINES
10733
10734                                   ;POWER DOWN ROUTINE
10735  054052  012737  054200  000024  $PWRDN: MOV     #$ILLUP,@#PWRVEC ;;SET FOR FAST UP
10736  054060  012737  000340  000026          MOV     #340,@#PWRVEC+2 ;;PRIO:7
10737  054066  010046                          MOV     R0,-(SP)        ;;PUSH R0 ON STACK
10738  054070  010146                          MOV     R1,-(SP)        ;;PUSH R1 ON STACK
10739  054072  010246                          MOV     R2,-(SP)        ;;PUSH R2 ON STACK
10740  054074  010346                          MOV     R3,-(SP)        ;;PUSH R3 ON STACK
10741  054076  010446                          MOV     R4,-(SP)        ;;PUSH R4 ON STACK
10742  054100  010546                          MOV     R5,-(SP)        ;;PUSH R5 ON STACK
10743  054102  010667  000076                  MOV     SP,$SAVR6       ;;SAVE SP
10744  054106  012737  054120  000024          MOV     #$PWRUP,@#PWRVEC ;;SET UP VECTOR
10745  054114  000000                          HALT
10746  054116  000776                          BR      .-2             ;;HANG UP
10747
10748                                   ;POWER UP ROUTINE
10749  054120  016706  000060          $PWRUP: MOV     $SAVR6,SP       ;;GET SP
10750  054124  005067  000054                  CLR     $SAVR6          ;;WAIT LOOP FOR THE TTY
10751  054130  005267  000050          1$:     INC     $SAVR6          ;;WAIT FOR THE INC
10752  054134  001375                          BNE     1$              ;;OF  WORD
10753  054136  012605                          MOV     (SP)+,R5        ;;POP STACK INTO R5
10754  054140  012604                          MOV     (SP)+,R4        ;;POP STACK INTO R4
10755  054142  012603                          MOV     (SP)+,R3        ;;POP STACK INTO R3
10756  054144  012602                          MOV     (SP)+,R2        ;;POP STACK INTO R2
10757  054146  012601                          MOV     (SP)+,R1        ;;POP STACK INTO R1
10758  054150  012600                          MOV     (SP)+,R0        ;;POP STACK INTO R0
10759  054152  012737  054052  000024          MOV     #$PWRDN,@#PWRVEC ;;SET UP THE POWER DOWN VECTOR
10760  054160  012737  000340  000026          MOV     #340,@#PWRVEC+2 ;;PRIO:7
10761  054166  104400                          TYPE                    ;REPORT THE POWER FAILURE
10762  054170  054206          $PWRMG: .WORD   $POWER                  ;;POWER FAIL MESSAGE POINTER
10763  054172  012716                  MOV     (PC)+,(SP)              ;;RESTART AT BEGIN
10764  054174  004220          $PWRAD: .WORD   BEGIN                   ;;RESTART ADDRESS
10765  054176  000002                  RTI
10766  054200  000000          $ILLUP: HALT                            ;;THE POWER UP SEQUENCE WAS STARTED
10767  054202  000776                  BR      .-2                     ;; BEFORE THE POWER DOWN WAS COMPLETE
10768  054204  000000          $SAVR6: 0                               ;;PUT THE SP HERE
10769  054206  005015  047520  042527 $POWER: .ASCIZ  <15><12>"POWER"
10770  054214  000122
10771                                           .EVEN
```

```
10772                                   ;*****************************************************************
10773                                   ;
10774                                   ;ERROR AND MESSAGE TABLE CONDIMENTS
10775                                   ;
10776                                   ;*****************************************************************
10777
10778
10779
10780
10781  054216  051127  047117  020107  EM1:    .ASCIZ  /WRONG DATA IN READING OR WRITING HARDWARE REGISTER/
10782  054224  040504  040524  044440
10783  054232  020116  042522  042101
10784  054240  047111  020107  051117
10785  054246  053440  044522  044524
10786  054254  043516  044040  051101
10787  054262  053504  051101  020105
10788  054270  042522  044507  052123
10789  054276  051105     000
10790  054301     105  051122  051117  EM2:    .ASCIZ  /ERROR ON  DATA COMMAND/
10791  054306  047440  020116  042040
10792  054314  052101  020101  047503
10793  054322  046515  047101  000104
10794  054330  051105  047522  020122  EM6:    .ASCIZ  /ERROR ON WRITE HEADER AND DATA/
10795  054336  047117  053440  044522
10796  054344  042524  044040  040505
10797  054352  042504  020122  047101
10798  054360  020104  040504  040524
10799  054366     000
10800  054367     103  047117  051124  EM11:   .ASCIZ  /CONTROLLER OR DRIVE STATUS/
10801  054374  046117  042514  020122
10802  054402  051117  042040  044522
10803  054410  042526  051440  040524
10804  054416  052524  000123
10805  054422  042522  044507  052123  EM14:   .ASCIZ  /REGISTER FAILED/
10806  054430  051105  043040  044501
10807  054436  042514  000104
10808  054442  050123  041505  043111  EM15:   .ASCIZ  /SPECIFIED REGISTER NON EXISTANT SO ABORT/
10809  054450  042511  020104  042522
10810  054456  044507  052123  051105
10811  054464  047040  047117  042440
10812  054472  044530  052123  047101
10813  054500  020124  047523  040440
10814  054506  047502  052122     000
10815  054513     127  044501  020124  EM16:   .ASCIZ  /WAIT LOOP FAILED/
10816  054520  047514  050117  043040
10817  054526  044501  042514  000104
10818  054534  051127  052111  020105  EM17:   .ASCIZ  /WRITE CHECK FAILING/
10819  054542  044103  041505  020113
10820  054550  040506  046111  047111
10821  054556  000107
10822  054560  042522  044507  052123  EM20:   .ASCIZ  /REGISTER FAILING/
10823  054566  051105  043040  044501
10824  054574  044514  043516     000
10825  054601     111  052116  051105  EM21:   .ASCIZ  /INTERRUPT FAILING/
```

```
10826  054606  052522  052120  043040
10827  054614  044501  044514  043516
10828  054622     000
10829  054623     105  051122  051117  EM22:   .ASCII  /ER' R ON DRIVES PRESENT -/<15><12>
10830  054630  047440  020116  051104
10831  054636  053111  051505  050040
10832  054644  042522  042523  052116
10833  054652  026440  005015
10834  054656  044124  020105  047125          .ASCII  /THE UNIT NO'S FOUND BY SETTING RHAS USING RHER1/<15><12>
10835  054664  052111  047040  023517
10836  054672  020123  047506  047125
10837  054700  020104  054502  051440
10838  054706  052105  044524  043516
10839  054714  051040  040510  020123
10840  054722  051525  047111  020107
10841  054730  044122  051105  006461
10842  054736     012
10843  054737     050  032124  020051          .ASCII  /(T4) DO NOT AG'EE WITH THE UNIT NO'S FOUND/<15><12>
10844  054744  047504  047040  052117
10845  054752  040440  051107  042505
10846  054760  053440  052111  020110
10847  054766  044124  020105  047125
10848  054774  052111  047040  023517
10849  055002  020123  047506  047125
10850  055010  006504     012
10851  055013     102  020131  042523          .ASCII  /BY SETTING RHCS2 - 'NED' (BIT #12)/<15><12><15><12>
10852  055020  052124  047111  020107
10853  055026  044122  051503  020062
10854  055034  020055  047047  042105
10855  055042  020047  041050  052111
10856  055050  021440  031061  006451
10857  055056  006412     012
10858  055061     116  052117  035105          .ASCII  /NOTE: ON DUAL PORT SYSTEM, A DRIVE ON OTHER PORT WILL /<15><12>
10859  055066  047440  020116  052504
10860  055074  046101  050040  051117
10861  055102  020124  054523  052123
10862  055110  046505  020054  020101
10863  055116  051104  053111  020105
10864  055124  047117  047440  044124
10865  055132  051105  050040  051117
10866  055140  020124  044527  046114
10867  055146  006440     012
10868  055151     116  052117  043440          .ASCII  /NOT GIVE 'NED', BUT WILL GIVE RHAS RESPONSES/<15><12>
10869  055156  053111  020105  047047
10870  055164  042105  026047  041040
10871  055172  052125  053440  046111
10872  055200  020114  044507  042526
10873  055206  051040  040510  020123
10874  055214  042522  050123  047117
10875  055222  042523  006523     012
10876  055227     110  047105  042503          .ASCIZ  /HENCE THERE WILL BE AN EXTRA DRIVE/
10877  055234  052040  042510  042522
10878  055242  053440  046111  020114
10879  055250  042502  040440  020116
```

```
10880  055256  054105  051124  020101
10881  055264  051104  053111  000105
10882  055272  047514  045517  040440  EM24:   .ASCIZ  /LOOK AHEAD REGISTER AT THE BEGINNING OF SECTOR IS IN ERROR/
10883  055300  042510  042101  051040
10884  055306  043505  051511  042524
10885  055314  020122  052101  052040
10886  055322  042510  041040  043505
10887  055330  047111  044516  043516
10888  055336  047440  020106  042523
10889  055344  052103  051117  044440
10890  055352  020123  047111  042440
10891  055360  051122  051117     000
10892  055365     114  047517  020113  EM25:   .ASCIZ  /LOOK AHEAD REGISTER IS IN ERROR/
10893  055372  044101  040505  020104
10894  055400  042522  044507  052123
10895  055406  051105  044440  020123
10896  055414  047111  042440  051122
10897  055422  051117     000
10898  055425     103  051125  042522  EM30:   .ASCII  /CURRENT CYLINDER DOES NOT MATCH DESIRED CYLINDER REGIHSTER/<15><12>
10899  055432  052116  041440  046131
10900  055440  047111  042504  020122
10901  055446  047504  051505  047040
10902  055454  052117  046440  052101
10903  055462  044103  042040  051505
10904  055470  051111  042105  041440
10905  055476  046131  047111  042504
10906  055504  020122  042522  044507
10907  055512  051510  042524  006522
10908  055520     012
10909  055521     101  052106  051105          .ASCIZ  /AFTER A SEEK AND INIT/
10910  055526  040440  051440  042505
10911  055534  020113  047101  020104
10912  055542  047111  052111     000
10913  055547     105  041503  043440  EM31:   .ASCII  /ECC GENERATED IS INCORRECT/<15><12>
10914  055554  047105  051105  052101
10915  055562  042105  044440  020123
10916  055570  047111  047503  051122
10917  055576  041505  006524     012
10918  055603     105  042526  054522          .ASCIZ  /EVERY WORD ON THIS SECTOR IS THAT GIVEN IN "DATA USED"/
10919  055610  053440  051117  020104
10920  055616  047117  052040  044510
10921  055624  020123  042523  052103
10922  055632  051117  044440  020123
10923  055640  044124  052101  043440
10924  055646  053111  047105  044440
10925  055654  020116  042042  052101
10926  055662  020101  051525  042105
10927  055670  000042
10928  055672  047117  051040  040305  EM32:   .ASCII  /ON READ COMMAND, AFTER DATA AND ECC HAVE BEEN READ,/<15><12>
10929  055700  020104  047503  046515
10930  055706  047101  026104  040440
10931  055714  052106  051105  042040
10932  055722  052101  020101  047101
10933  055730  020104  041505  020103
```

```
10934  055736  040510  042526  041040
10935  055744  042505  020116  042522
10936  055752  042101  006454     012
10937  055757     105  041503  051040          .ASCII  /ECC REGISTERS OR RHER1 ARE IN ERROR/<15><12>
10938  055764  043505  051511  042524
10939  055772  051522  047440  020122
10940  056000  044122  051105  020061
10941  056006  051101  020105  047111
10942  056014  042440  051122  051117
10943  056022  005015
10944  056024  047117  054514  046040          .ASCII  /ONLY LOWER 11 BITS OF PATTERN REG. CAN BE READ/<15><12>
10945  056032  053517  051105  030440
10946  056040  020061  044502  051524
10947  056046  047440  020106  040520
10948  056054  052124  051105  020116
10949  056062  042522  027107  041440
10950  056070  047101  041040  020105
10951  056076  042522  042101  005015
10952  056104  044124  051511  051440          .ASCIZ  /THIS SHOULD MATCH LOWER 11 BITS OF GOOD ECC1/
10953  056112  047510  046125  020104
10954  056120  040515  041524  020110
10955  056126  047514  042527  020122
10956  056134  030461  041040  052111
10957  056142  020123  043117  043440
10958  056150  047517  020104  041505
10959  056156  030503     000
10960  056161     110  043511  020110  EM33:   .ASCIZ  /HIGH COUNT BIT NOT SET AFTER 38859 CLOCKS/
10961  056166  047503  047125  020124
10962  056174  044502  020124  047516
10963  056202  020124  042523  020124
10964  056210  043101  042524  020122
10965  056216  034063  032470  020071
10966  056224  046103  041517  051513
10967  056232     000
10968  056233     132  051105  020117  EM34:   .ASCIZ  /ZERO DETECT BIT NOT HIGH WHEN 32 BIT ECC REG. HAS 21 ZEROS/
10969  056240  042504  042524  052103
10970  056246  041040  052111  047040
10971  056254  052117  044040  043511
10972  056262  020110  044127  047105
10973  056270  031440  020062  044502
10974  056276  020124  041505  020103
10975  056304  042522  027107  044040
10976  056312  051501  031040  020061
10977  056320  042532  047522  000123
10978  056326  047520  044523  044524  EM35:   .ASCII  /POSITION REGISTER OR 11 BITS OF PATTERN REGISTER INCORRECT/<15><12>
10979  056334  047117  051040  043505
10980  056342  051511  042524  020122
10981  056350  051117  030440  020061
10982  056356  044502  051524  047440
10983  056364  020106  040520  052124
10984  056372  051105  020116  042522
10985  056400  044507  052123  051105
10986  056406  044440  041516  051117
10987  056414  042522  052103  005015
```

```
10988  056422  047514  042527  020122          .ASCII  /LOWER 11 BITS OF PATTERN REGISTER SHOULD MATCH LOWER/<15><12>
10989  056430  030461  041040  052111
10990  056436  020123  043117  050040
10991  056444  052101  042524  047122
10992  056452  051040  043505  051511
10993  056460  042524  020122  044123
10994  056466  052517  042114  046440
10995  056474  052101  044103  046040
10996  056502  053517  051105  005015
10997  056510  030461  041040  052111          .ASCII  /11 BITS OF GOOD ECC1/<15><12>
10998  056516  020123  043117  043440
10999  056524  047517  020104  041505
11000  056532  030503  005015
11001  056536  040504  020124  047105          .ASCIZ  /DAT ENVLOP GOOD POSITION AND N-CODE ZEROS ARE IN OCTAL/
11002  056544  046126  050117  043440
11003  056552  047517  020104  047520
11004  056560  044523  044524  047117
11005  056566  040440  042116  047040
11006  056574  041455  042117  020105
11007  056602  042532  047522  020123
11008  056610  051101  020105  047111
11009  056616  047440  052103  046101
11010  056624     000
11011  056625     117  020116  042522  EM36:   .ASCIZ  /ON READ COMMAND WITH NON-CORRECTABLE ERROR 'DCK' AND 'ECH' SHOULD BE SE
11012  056632  042101  041440  046517
11013  056640  040515  042116  053440
11014  056646  052111  020110  047516
11015  056654  026516  047503  051122
11016  056662  041505  040524  046102
11017  056670  020105  051105  047522
11018  056676  020122  042047  045503
11019  056704  020047  047101  020104
11020  056712  042447  044103  020047
11021  056720  044123  052517  042114
11022  056726  041040  020105  042523
11023  056734  000124
11024  056736  051120  043517  040522  EM37:   .ASCII  /PROGRAM ERROR BIT #10 IN RHCS2 DID NOT SET/<15><12>
11025  056744  020115  051105  047522
11026  056752  020122  044502  020124
11027  056760  030443  020060  047111
11028  056766  051040  041510  031123
11029  056774  042040  042111  047040
11030  057002  052117  051440  052105
11031  057010  005015
11032  057012  043111  050040  051517          .ASCIZ  /IF POSITION REGISTER =10040 OR 10041, IT IS GOOD/
11033  057020  052111  047511  020116
11034  057026  042522  044507  052123
11035  057034  051105  036440  030061
11036  057042  032060  020060  051117
11037  057050  030440  030060  030464
11038  057056  020054  052111  044440
11039  057064  020123  047507  042117
11040  057072     000
11041
```

F08


```
 11042  057073     122  053510  020103  EM40:   .ASCII  /RHWC DID NOT = 0 UPON COMPLETION OF READ/<15><12>
 11043  057100  044504  020104  047516
 11044  057106  020124  020075  020060
 11045  057114  050125  047117  041440
 11046  057122  046517  046120  052105
 11047  057130  047511  020116  043117
 11048  057136  051040  040505  006504
 11049  057144     012
 11050  057145     117  020122  051127          .ASCIZ  /OR WRITE HEADER AND DATA/
 11051  057152  052111  020105  042510
 11052  057160  042101  051105  040440
 11053  057166  042116  042040  052101
 11054  057174  000101
```

```
11055
11056  057176  040506  040524  020114  CPHALT: .ASCII  /FATAL ERROR - SEE DOCUMENT LISTING/<15><12>
11057  057204  051105  047522  020122
11058  057212  020055  042523  020105
11059  057220  047504  052503  042515
11060  057226  052116  046040  051511
11061  057234  044524  043516  005015
11062  057242  006440  103412  177777          .ASCII  / /<15><12><207><377><377><207><377><377><207><377><377>
11063  057250  177607  103777  177777
11064  057256  044124  020105  047503          .ASCII  /THE CONTROLLER OR DEVICE HAS GONE OFFLINE, LOST/<15><12>
11065  057264  052116  047522  046114
11066  057272  051105  047440  020122
11067  057300  042504  044526  042503
11068  057306  044040  051501  043440
11069  057314  047117  020105  043117
11070  057322  046106  047111  026105
11071  057330  046040  051517  006524
11072  057336     012
11073  057337     047  042522  042101          .ASCII  /'READY', BECOME UNAVAILABLE, OR HAS STATUS BITS/<15><12>
11074  057344  023531  020054  042502
11075  057352  047503  042515  052440
11076  057360  040516  040526  046111
11077  057366  041101  042514  020054
11078  057374  051117  044040  051501
11079  057402  051440  040524  052524
11080  057410  020123  044502  051524
11081  057416  005015
11082  057420  044127  041511  020110          .ASCIZ  /WHICH CANNOT BE CLEARED/
11083  057426  040503  047116  052117
11084  057434  041040  020105  046103
11085  057442  040505  042522  000104
11086
11087  057450  020040  020040  020040  SPACE8: .ASCII  /     /
11088  057456  020040     000          SPACE2: .ASCIZ  / /
11089
11090
11091  057461     120  020103  020040  DH1:    .ASCII  /PC      TEST    REG.    GOOD    RECEIVED/<15><12>
11092  057466  020040  052040  051505
11093  057474  020124  020040  051040
11094  057502  043505  020056  020040
11095  057510  043440  047517  020104
11096  057516  020040  051040  041505
11097  057524  044505  042526  006504
11098  057532     012
11099  057533     040  020040  020040          .ASCIZ  /       NO      ADDR.   DATA    DATA    /
11100  057540  020040  047040  020117
11101  057546  020040  020040  040440
11102  057554  042104  027122  020040
11103  057562  042040  052101  020101
11104  057570  020040  042040  052101
11105  057576  020101  020040  000040
11106  057604  041520  020040  020040  DH2:    .ASCII  /PC      TEST    WORD    GOOD    BAD     /<15><12>
11107  057612  020040  042524  052123
11108  057620  020040  020040  047527
```

```
11109  057626  042122  020040  020040
11110  057634  047507  042117  020040
11111  057642  020040  040502  020104
11112  057650  020040  020040  005015
11113  057656  020040  020040  020040          .ASCIZ  /       NO      NO      DATA    DATA    /
11114  057664  020040  047516  020040
11115  057672  020040  020040  047516
11116  057700  020040  020040  020040
11117  057706  040504  040524  020040
11118  057714  020040  040504  040524
11119  057722  020040  020040     000
11120  057727     120  020103  020040  DH11:   .ASCII  /PC     TEST    FAILING CONT.   CONT.   CONT.   CONT.   /<15><12>
11121  057734  020040  052040  051505
11122  057742  020124  020040  043040
11123  057750  044501  044514  043516
11124  057756  041440  047117  027124
11125  057764  020040  041440  047117
11126  057772  027124  020040  041440
11127  060000  047117  027124  020040
11128  060006  041440  047117  027124
11129  060014  020040  006440     012
11130  060021     040  020040  020040          .ASCIZ  /       NO      REG ADR RHCS1   RHCS2   RHDS1   RHER1/
11131  060026  020040  047040  020117
11132  060034  020040  020040  051040
11133  060042  043505  040440  051104
11134  060050  051040  041510  030523
11135  060056  020040  051040  041510
11136  060064  031123  020040  051040
11137  060072  042110  030523  020040
11138  060100  051040  042510  030522
11139  060106     000
11140  060107     120  020103  020040  DH14:   .ASCII  /PC     TEST    FAILING CONT.   CONT.   CONT.   CONT.   CONT.   /<15><1
11141  060114  020040  052040  051505
11142  060122  020124  020040  043040
11143  060130  044501  044514  043516
11144  060136  041440  047117  027124
11145  060144  020040  041440  047117
11146  060152  027124  020040  041440
11147  060160  047117  027124  020040
11148  060166  041440  047117  027124
11149  060174  020040  041440  047117
11150  060202  027124  020040  006440
11151  060210     012
11152  060211     040  020040  020040          .ASCIZ  /       NO      REG ADR BAD REG RHCS1   RHCS2   RHDS1   RHER1/
11153  060216  020040  047040  020117
11154  060224  020040  020040  051040
11155  060232  043505  040440  051104
11156  060240  041040  042101  051040
11157  060246  043505  051040  041510
11158  060254  030523  020040  051040
11159  060262  041510  031123  020040
11160  060270  051040  042110  030523
11161  060276  020040  051040  042510
11162  060304  030522     000
```

```
11163  060307    120  020103  020040  DH15:  .ASCIZ  /PC      TEST    REG ADR /
11164  060314  020040  052040  051505
11165  060322  020124  020040  051040
11166  060330  043505  040440  051104
11167  060336  000040
11168  060340  041520  020040  020040  DH16:  .ASCII  /PC      TEST    WAIT    BIT     REG     REG     /<15><12>
11169  060346  020040  042524  052123
11170  060354  020040  020040  040527
11171  060362  052111  020040  020040
11172  060370  044502  020124  020040
11173  060376  020040  042522  020107
11174  060404  020040  020040  042522
11175  060412  020107  020040  020040
11176  060420  005015
11177  060422  020040  020040  020040          .ASCIZ  /       NO      PC      WANTED  ADDRESS CONT.   /
11178  060430  020040  047516  020040
11179  060436  020040  020040  041520
11180  060444  020040  020040  020040
11181  060452  040527  052116  042105
11182  060460  020040  042101  051104
11183  060466  051505  020123  047503
11184  060474  052116  020056  020040
11185  060502     000
11186  060503    120  020103  020040  DH17:  .ASCII  /PC      TEST    CONT.   CONT.   CONT.   CONT.   CONT.   /<15><12>
11187  060510  020040  052040  051505
11188  060516  020124  020040  041440
11189  060524  047117  027124  020040
11190  060532  041440  047117  027124
11191  060540  020040  041440  047117
11192  060546  027124  020040  041440
11193  060554  047117  027124  020040
11194  060562  041440  047117  027124
11195  060570  020040  006440     012
11196  060575     040  020040  020040          .ASCIZ  /       NO      RHBA    RHDB    RHMC    RHCS1   RHCS2   /
11197  060602  020040  047040  020117
11198  060610  020040  020040  051040
11199  060616  041110  020101  020040
11200  060624  051040  042110  020102
11201  060632  020040  051040  053510
11202  060640  020103  020040  051040
11203  060646  041510  030523  020040
11204  060654  051040  041510  031123
11205  060662  020040  000040
11206  060666  041520  020040  020040  DH20:  .ASCII  /PC      TEST    CONT    CONT    CONT    CONT    CONT    /<15><12>
11207  060674  020040  042524  052123
11208  060702  020040  020040  047503
11209  060710  052116  020040  020040
11210  060716  047503  052116  020040
11211  060724  020040  047503  052116
11212  060732  020040  020040  047503
11213  060740  052116  020040  020040
11214  060746  047503  052116  020040
11215  060754  020040  005015
11216  060760  020040  020040  020040          .ASCIZ  /       NO      RHER1   RHER2   RHER3   RHAS    RHDS1   /
```

```
11217  060766  020040  047516  020040
11218  060774  020040  020040  044122
11219  061002  051105  020061  020040
11220  061010  044122  051105  020062
11221  061016  020040  044122  051105
11222  061024  020063  020040  044122
11223  061032  051501  020040  020040
11224  061040  044122  051504  020061
11225  061046  020040     000
11226  061051     120  020103  020040  DH21:   .ASCII  /PC      TEST    CONT    CONT    CONT    /<15><12>
11227  061056  020040  052040  051505
11228  061064  020124  020040  041440
11229  061072  047117  020124  020040
11230  061100  041440  047117  020124
11231  061106  020040  041440  047117
11232  061114  020124  020040  006440
11233  061122     012
11234  061123     040  020040  020040          .ASCIZ  /       NO      RHCS1   RHAS    RHDS1/
11235  061130  020040  047040  020117
11236  061136  020040  020040  051040
11237  061144  041510  030523  020040
11238  061152  051040  040510  030123
11239  061160  020040  051040  042110
11240  061166  030523     000
11241  061171     120  020103  020040  DH22:   .ASCII  /PC      TEST       RHAS    RHCS2/<15><12>
11242  061176  020040  052040  051505
11243  061204  020124  020040  020040
11244  061212  020040  044122  051501
11245  061220  020040  020040  044122
11246  061226  051503  006462     012
11247  061233     040  020040  020040          .ASCIZ  /       NO         UNITS   UNITS/
11248  061240  020040  047040  020117
11249  061246  020040  020040  020040
11250  061254  020040  047125  052111
11251  061262  020123  020040  047125
11252  061270  052111  000123
11253  061274  041520  020040  020040  DH24:   .ASCII  /PC      TEST    RHDST   BAD     GOOD    SECTOR  SECTOR  /<15><12>
11254  061302  020040  042524  052123
11255  061310  020040  020040  044122
11256  061316  051504  020124  020040
11257  061324  040502  020104  020040
11258  061332  020040  047507  042117
11259  061340  020040  020040  042523
11260  061346  052103  051117  020040
11261  061354  042523  052103  051117
11262  061362  020040  005015
11263  061366  020040  020040  020040          .ASCIZ  /       NO      CONT.   RHLA    RHLA    NO      CLOCK   /
11264  061374  020040  047516  020040
11265  061402  020040  020040  047503
11266  061410  052116  020056  020040
11267  061416  044122  040514  020040
11268  061424  020040  044122  040514
11269  061432  020040  020040  047516
11270  061440  020040  020040  020040
```

```
11271  061446  046103  041517  020113
11272  061454  020040     000
11273  061457     120  020103  020040  DH26:   .ASCII  /PC     TEST   PC OF  FAILING CONT.  CONT.  CONT.  CONT.  /<15><1
11274  061464  020040  052040  051505
11275  061472  020124  020040  050040
11276  061500  020103  043117  020040
11277  061506  043040  044501  044514
11278  061514  043516  041440  047117
11279  061522  027124  020040  041440
11280  061530  047117  027124  020040
11281  061536  041440  047117  027124
11282  061544  020040  041440  047117
11283  061552  027124  020040  006440
11284  061560     012
11285  061561     040  020040  020040          .ASCIZ  /      NO     JSR    REG ADD RHCS1  RHCS2  RHDS1  RHER1  /
11286  061566  020040  047040  020117
11287  061574  020040  020040  045040
11288  061602  051123  020040  020040
11289  061610  051040  043505  040440
11290  061616  042104  051040  041510
11291  061624  030523  020040  051040
11292  061632  041510  031123  020040
11293  061640  051040  042110  030523
11294  061646  020040  051040  042510
11295  061654  030522  020040  000040
11296  061662  041520  020040  020040  DH27:   .ASCII  /PC     TEST   PC OF  FAILING GOOD   RECEIVED/<15><12>
11297  061670  020040  042524  052123
11298  061676  020040  020040  041520
11299  061704  047440  020106  020040
11300  061712  040506  046111  047111
11301  061720  020107  047507  042117
11302  061726  020040  020040  042522
11303  061734  042503  053111  042105
11304  061742  005015
11305  061744  020040  020040  020040          .ASCIZ  /      NO     JSR    REG    DATA   DATA   /
11306  061752  020040  047516  020040
11307  061760  020040  020040  051512
11308  061766  020122  020040  020040
11309  061774  042522  020107  020040
11310  062002  020040  040504  040524
11311  062010  020040  020040  040504
11312  062016  040524  020040  020040
11313  062024     000
11314  062025     120  020103  020040  DH30:   .ASCII  /PC     TEST   PC OF  REG.    GOOD   BAD    /<15><12>
11315  062032  020040  052040  051505
11316  062040  020124  020040  050040
11317  062046  020103  043117  020040
11318  062054  051040  043505  020056
11319  062062  020040  043440  047517
11320  062070  020104  020040  041040
11321  062076  042101  020040  020040
11322  062104  006440     012
11323  062107     040  020040  020040          .ASCIZ  /      NO     JSR    ADDRESS DATA   DATA   /
11324  062114  020040  047040  020117
```

```
11325  062122  020040  020040  045040
11326  062130  051123  020040  020040
11327  062136  040440  042104  042522
11328  062144  051523  042040  052101
11329  062152  020101  020040  042040
11330  062160  052101  020101  020040
11331  062166  000040
11332  062170  041520  020040  020040  DH31:   .ASCII  /PC      TEST    WORD    GOOD    BAD     CONT.   CONT.   CONT    /<15><1
11333  062176  020040  042524  052123
11334  062204  020040  020040  047527
11335  062212  042122  020040  020040
11336  062220  047507  042117  020040
11337  062226  020040  040502  020104
11338  062234  020040  020040  047503
11339  062242  052116  020056  020040
11340  062250  047503  052116  020056
11341  062256  020040  047503  052116
11342  062264  020040  020040  005015
11343  062272  020040  020040  020040          .ASCII  /       NO      NO      DATA    DATA    RHCS1   RHDS1   RHER1   /
11344  062300  020040  047516  020040
11345  062306  020040  020040  047516
11346  062314  020040  020040  020040
11347  062322  040504  040524  020040
11348  062330  020040  040504  040524
11349  062336  020040  020040  044122
11350  062344  051503  020061  020040
11351  062352  044122  051504  020061
11352  062360  020040  044122  051105
11353  062366  020061  020040
11354  062372  041520  020040  020040  DH32:   .ASCII  /PC      TEST    PC OF   WORD    GOOD    BAD     CONT.   CONT.   CONT.
11355  062400  020040  042524  052123
11356  062406  020040  020040  041520
11357  062414  047440  020106  020040
11358  062422  047527  042122  020040
11359  062430  020040  047507  042117
11360  062436  020040  020040  040502
11361  062444  020104  020040  020040
11362  062452  047503  052116  020056
11363  062460  020040  047503  052116
11364  062466  020056  020040  047503
11365  062474  052116  020056  020040
11366  062502  005015
11367  062504  020040  020040  020040          .ASCIZ  /       NO      JSR     NO      DATA    DATA    RHCS1   RHDS1   RHER1
11368  062512  020040  047516  020040
11369  062520  020040  020040  051512
11370  062526  020122  020040  020040
11371  062534  047516  020040  020040
11372  062542  020040  040504  040524
11373  062550  020040  020040  040504
11374  062556  040524  020040  020040
11375  062564  044122  051503  020061
11376  062572  020040  044122  051504
11377  062600  020061  020040  044122
11378  062606  051105  020061  020040
```

```
11379  062614     000
11380  062615     120  020103  020040  DH33:  .ASCII  /PC      TEST    PC OF   WORD    GOOD    CONT.   CONT.   CONT.   /<15><1
11381  062622  020040  052040  051505
11382  062630  020124  020040  050040
11383  062636  020103  043117  020040
11384  062644  053440  051117  020104
11385  062652  020040  043440  047517
11386  062660  020104  020040  041440
11387  062666  047117  027124  020040
11388  062674  041440  047117  027124
11389  062702  020040  041440  047117
11390  062710  027124  020040  006440
11391  062716     012
11392  062717     040  020040  020040         .ASCIZ  /       NO      JSR     NO      DATA    RHCS1   RHDS1   RHER1   /
11393  062724  020040  047040  020117
11394  062732  020040  020040  045040
11395  062740  051123  020040  020040
11396  062746  047040  020117  020040
11397  062754  020040  042040  052101
11398  062762  020101  020040  051040
11399  062770  041510  030523  020040
11400  062776  051040  042110  030523
11401  063004  020040  051040  042510
11402  063012  030522  020040  000040
11403  063020  041520  020040  020040  DH34:  .ASCII  /PC      TEST    GOOD    GOOD    WRITTEN WRITTEN DATA    /<15><12>
11404  063026  020040  042524  052123
11405  063034  020040  020040  047507
11406  063042  042117  020040  020040
11407  063050  047507  042117  020040
11408  063056  020040  051127  052111
11409  063064  042524  020116  051127
11410  063072  052111  042524  020116
11411  063100  040504  040524  020040
11412  063106  020040  005015
11413  063112  020040  020040  020040         .ASCIZ  /       NO      ECC1    ECC2    ECC1    ECC2    USED    /
11414  063120  020040  047516  020040
11415  063126  020040  020040  041505
11416  063134  030503  020040  020040
11417  063142  041505  031103  020040
11418  063150  020040  041505  030503
11419  063156  020040  020040  041505
11420  063164  031103  020040  020040
11421  063172  051525  042105  020040
11422  063200  020040     000
11423  063203     120  020103  020040  DH35:  .ASCII  /PC      TEST    GOOD    GOOD    PATTERN POSITON GOOD    RHER1   /<15><1
11424  063210  020040  052040  051505
11425  063216  020124  020040  043440
11426  063224  047517  020104  020040
11427  063232  043440  047517  020104
11428  063240  020040  050040  052101
11429  063246  042524  047122  050040
11430  063254  051517  052111  047117
11431  063262  043440  047517  020104
11432  063270  020040  051040  042510
```

```
11433  063276  030522  020040  006440
11434  063304     012
11435  063305     040  020040  020040          .ASCIZ  /       NO      ECC1    ECC2    REG.REG.        POSITON REG.    /
11436  063312  020040  047040  020117
11437  063320  020040  020040  042440
11438  063326  041503  020061  020040
11439  063334  042440  041503  020062
11440  063342  020040  051040  043505
11441  063350  051056  043505  020056
11442  063356  020040  020040  020040
11443  063364  050040  051517  052111
11444  063372  047117  051040  043505
11445  063400  020056  020040  000040
11446  063406  041520  020040  020040  DH36:   .ASCII  /PC      TEST    PC OF   RHMR    POSITON PATTERN /<15><12>
11447  063414  020040  042524  052123
11448  063422  020040  020040  041520
11449  063430  047440  020106  020040
11450  063436  044122  051115  020040
11451  063444  020040  047520  044523
11452  063452  047524  020116  040520
11453  063460  052124  051105  020116
11454  063466  005015
11455  063470  020040  020040  020040          .ASCIZ  /       NO      JSR     CONT.   REG.    REG./
11456  063476  020040  047516  020040
11457  063504  020040  020040  051512
11458  063512  020122  020040  020040
11459  063520  047503  052116  020056
11460  063526  020040  042522  027107
11461  063534  020040  020040  042522
11462  063542  027107     000
11463  063545     120  020103  020040  DH37:   .ASCII  /PC      TEST    POSITON POSITON GOOD    GOOD    PATTERN DATA    N-CODE/
11464  063552  020040  052040  051505
11465  063560  020124  020040  050040
11466  063566  051517  052111  047117
11467  063574  050040  051517  052111
11468  063602  047117  043440  047517
11469  063610  020104  020040  043440
11470  063616  047517  020104  020040
11471  063624  050040  052101  042524
11472  063632  047122  042040  052101
11473  063640  020101  020040  047040
11474  063646  041455  042117  006505
11475  063654     012
11476  063655     040  020040  020040          .ASCIZ  /       NO      ECC     GOOD    ECC1    ECC2    ECC     ENVLOPE ZEROS/
11477  063662  020040  047040  020117
11478  063670  020040  020040  042440
11479  063676  041503  020040  020040
11480  063704  043440  047517  020104
11481  063712  020040  042440  041503
11482  063720  020061  020040  042440
11483  063726  041503  020062  020040
11484  063734  042440  041503  020040
11485  063742  020040  042440  053116
11486  063750  047514  042520  055040
```

```
11487  063756  051105  051517     000
11488
11489  063763     120  020103  020040  DH40:   .ASCII  /PC      TEST    CONT./<15><12>
11490  063770  020040  052040  051505
11491  063776  020124  020040  041440
11492  064004  047117  027124  005015
[illegible]                            .ASCIZ  /       [illegible]
11494  064020  020040  047516  020040
11495  064026  020040  020040  044122
11496  064034  041527     000
11497
11498          064040                  .EVEN
11499
11500  064040  001116  002020  040172  DT1:    .WORD   SERRPC,TSTNM,REGADR,SGDDAT,SBDDAT,0
11501  064046  001124  001126  000000
11502  064054  001116  002020  045272  DT2:    .WORD   SERRPC,TSTNM,ERWORD,SGDDAT,0
11503  064062  001124  000000
11504  064066  001116  002020  045272  DT3:    .WORD   SERRPC,TSTNM,ERWORD,SGDDAT,SBDDAT,0
11505  064074  001124  001126  000000
11506
11507  064102  001116  002020  001122  DT11:   .WORD   SERRPC,TSTNM,SBDADR,CS1,CS2,DS1,ER1,0
11508  064110  001702  001700  001724
11509  064116  001704  000000
11510  064122  001116  002020  001122  DT14:   .WORD   SERRPC,TSTNM,SBDADR,SBDDAT,CS1,CS2,DS1,ER1,0
11511  064130  001126  001702  001700
11512  064136  001724  001704  000000
11513  064144  001116  002020  001172  DT15:   .WORD   SERRPC,TSTNM,STMP1,0
11514  064152  000000
11515  064154  001116  002020  001176  DT16:   .WORD   SERRPC,TSTNM,STMP3,STMP1,STMP0,SBDDAT,0
11516  064162  001172  001170  001126
11517  064170  000000
11518  064172  001116  002020  001676  DT17:   .WORD   SERRPC,TSTNM,BA,DB,WC,CS1,CS2,0
11519  064200  001672  001674  001702
11520  064206  001700  000000
11521
11522  064212  001116  002020  001704  DT20:   .WORD   SERRPC,TSTNM,ER1,ER2,ER3,AS,DS1,0
11523  064220  001710  001716  001720
11524  064226  001724  000000
11525  064232  001116  002020  001702  DT21:   .WORD   SERRPC,TSTNM,CS1,AS,DS1,0
11526  064240  001720  001724  000000
11527  064246  001116  002020  000000  DT22:   .WORD   SERRPC,TSTNM,0
11528  064254  001116  002020  001706  DT24:   .WORD   SERRPC,TSTNM,DST,SBDDAT,STMP1,STMP2,STMP3,0
11529  064262  001126  001172  001174
11530  064270  001176  000000
11531  064274  001116  002020  002002  DT26:   .WORD   SERRPC,TSTNM,PCJSR,SBDADR,CS1,CS2,DS1,ER1,0
11532  064302  001122  001702  001700
11533  064310  001724  001704  000000
11534  064316  001116  002020  002002  DT27:   .WORD   SERRPC,TSTNM,PCJSR,REGADR,SGDDAT,SBDDAT,0
11535  064324  040172  001124  001126
11536  064332  000000
11537
11538  064334  001116  002020  002002  DT30:   .WORD   SERRPC,TSTNM,PCJSR,REGADR,SGDDAT,SBDDAT,0
11539  064342  040172  001124  001126
11540  064350  000000
```

```
11541  064352  001116  002020  045272  DT31:   .WORD   SERRPC,TSTNM,ERWORD,SGDDAT,SBDDAT,CS1,DS1,ER1,0
11542  064360  001124  001126  001702
11543  064366  001724  001704  000000
11544  064374  001116  002020  002002  DT32:   .WORD   SERRPC,TSTNM,PCJSR,ERWORD,SGDDAT,SBDDAT,CS1,DS1,ER1,0
11545  064402  045272  001124  001126
11546  064410  001702  001724  001704
11547  064416  000000
11548  064420  001116  002020  002002  DT33:   .WORD   SERRPC,TSTNM,PCJSR,ERWORD,SGDDAT,CS1,DS1,ER1,0
11549  064426  045272  001124  001702
11550  064434  001724  001704  000000
11551  064442  001116  002020  043216  DT34:   .WORD   SERRPC,TSTNM,GECC1,GECC2,MECC1,MECC2,DISK,0
11552  064450  043220  050070  050072
11553  064456  047070  000000
11554  064462  001116  002020  043216  DT35:   .WORD   SERRPC,TSTNM,GECC1,GECC2,EC2,EC1,POSITI,ER1,0
11555  064470  043220  001734  001732
11556  064476  043230  001704  000000
11557  064504  001116  002020  002002  DT36:   .WORD   SERRPC,TSTNM,PCJSR,MR,EC1,EC2,0
11558  064512  001722  001732  001734
11559  064520  000000
11560  064522  001116  002020  001732  DT37:   .WORD   SERRPC,TSTNM,EC1,POSITI,GECC1,GECC2,EC2,DATENV,ZCODE,0
11561  064530  043230  043216  043220
11562  064536  001734  043234  043236
11563  064544  000000
11564
11565  064546  001116  002020  001126  DT40:   .WORD   SERRPC,TSTNM,SBDDAT,0
11566  064554  000000
11567
11568  064556  000  000  000  DF1:    .BYTE   0,0,0,0,0
11569  064561  000  000
11570  064563  000  000  001  DF2:    .BYTE   0,0,1,0
11571  064566  000
11572  064567  000  000  001  DF3:    .BYTE   0,0.1,0,0
11573  064572  000  000
11574
11575  064574  000  000  000  DF11:   .BYTE   0,0,0,0,0,0,0
11576  064577  000  000  000
11577  064602  000
11578  064603  000  000  000  DF14:   .BYTE   0,0,0,0,0,0,0,0
11579  064606  000  000  000
11580  064611  000  000
11581  064613  000  000  000  DF15:   .BYTE   0,0,0
11582  064616  000  000  000  DF16:   .BYTE   0,0,0,0,0
11583  064621  000  000
11584  064623  000  000  000  DF17:   .BYTE   0,0,0,0,0,0,0
11585  064626  000  000  000
11586  064631  000
11587
11588  064632  000  000  000  DF20:   .BYTE   0,0,0,0,0,0,0
11589  064635  000  000  000
11590  064640  000
11591  064641  000  000  000  DF21:   .BYTE   0,0,0,0,0
11592  064644  000  000
11593  064646  000  000  000  DF22:   .BYTE   0,0,0,0
11594  064651  000
```

```
11595  064652  000  000  000  DF24:  .BYTE  0,0,0,0,0,0,0
11596  064655  000  000  000
11597  064660  000
11598  064661  000  000  000  DF26:  .BYTE  0,0,0,0,0,0,0,0
11599  064664  000  000  000
11600  064667  000  000
11601  064671  000  000  000  DF27:  .BYTE  0,0,0,0,0,0
11602  064674  000  000  000
11603
11604  064677  000  000  000  DF30:  .BYTE  0,0,0,0,0,0
11605  064702  000  000  000
11606  064705  000  000  001  DF31:  .BYTE  0,0,1,0,0,0,0,0
11607  064710  000  000  000
11608  064713  000  000
11609  064715  000  000  000  DF32:  .BYTE  0,0,0,1,0,0,0,0,0
11610  064720  001  000  000
11611  064723  000  000  000
11612  064726  000  000  000  DF33:  .BYTE  0,0,0,1,0,0,0,0
11613  064731  001  000  000
11614  064734  000  000
11615  064736  000  000  000  DF34:  .BYTE  0,0,0,0,0,0,0
11616  064741  000  000  000
11617  064744  000
11618  064745  000  000  000  DF35:  .BYTE  0,0,0,0,0,0,0,0
11619  064750  000  000  000
11620  064753  000  000
11621  064755  000  000  000  DF36:  .BYTE  0,0,0,0,0,0
11622  064760  000  000  000
11623  064763  000  000  000  DF37:  .BYTE  0,0,0,0,0,0,0,0,0
11624  064766  000  000  000
11625  064771  000  000  000
11626
11627  064774  000  000  000  DF40:  .BYTE  0,0,0
11628          065000                .EVEN
11629
11630          000001                .END
```

DZRPSD.P11 CROSS REFERENCE TABLE

```
ACL    = 000040            943#
ACU    = 100000            895#
AOE    = 001000            811#  7369
AS       001720           1029#  3133   3201   5948   6006  11522  11525
ATA    = 100000            799#  2970   4541   5958   5978   7375   7518   7628   7734
ATABLE   004200           1117#  1462
ATTENT   002004           1061#  1460*  1462*  3405   4538   5936   6004   7373   7516   7626   7732
AT0    = 000001            833#
AT1    = 000002            834#
AT2    = 000004            835#
AT3    = 000010            836#
AT4    = 000020            837#
AT5    = 000040            838#
AT6    = 000100            839#
AT7    = 000200            840#
A16    = 000400            774#  3034   6187   6207
A17    = 001000            775#  3034   6187   6207
BA       00167%           1018#  6335  11518
BAI    = 00001(            744#  2806   2811   6088   6199   6215   6669   8681
BASECH   044146           1254   9082#
BEGIN    004220            174   1130#  10764
BEGIN2   004210            176   1128#
BITST    04017%           1279   1540   1565   1641   1665   1689   1771   1797   1822   1847   1868   7970#
BIT0   = 000001            147#  5943   8423   8458   8460
BIT00  = 000001            137#   147
BIT01  = 000002            136#   146
BIT02  = 000004            135#   145
BIT03  = 000010            134#   144
BIT04  = 000020            133#   143
BIT05  = 000040            132#   142   1748
BIT06  = 000100            131#   141   1748  10384
BIT07  = 000200            130#   140   1748
BIT08  = 000400            129#   139   1749  10211
BIT09  = 001000            128#   138   1727   1748  10219  10551
BIT1   = 000002            146#  8437
BIT10  = 002000            127#  10537
BIT11  = 004000            126#  10226
BIT12  = 010000            125#
BIT13  = 020000            124#  8453   8455  10544
BIT14  = 040000            123#  8430
BIT15  = 100000            122#  8425   8432   8439   8448   8450
BIT2   = 000004            145#
BIT3   = 000010            144#
BIT4   = 000020            143#
BIT5   = 000040            142#
BIT6   = 000100            141#
BIT7   = 000200            140#
BIT8   = 000400            139#
BIT9   = 001000            138#
BLT1     040222           7976#  7981
BLT2     040244           7976   7977   7983#
BLT3     040254           7984   7985#
BPTVEC= 000014              154#
CA       001714           1027#
```

DZRPSD.P11 CROSS REFERENCE TABLE

```
CC       001740          1037#
CHECKD=  104422          10728#
CHECKE   040746          4673    4783    4907    5135    5262    5388    6480    8141#
CHECKT   040556          3494    4394    4643    4756    4878    4997    5104    5231    5357    5475    5569    5919    6084
                         6195    6449    6577    6653    7234    7346    7489    7595    7701    8089#   8270    8370    8537
                        10728
CLAREA   040440          4681    4791    4795    4915    4919    5441    5463    5533    5538    5586    5591    5660    5672
                         5679    5691    5728    5740    5801    5821    5866    6048    6053    6101    6162    6278    6298
                         6357    6549    6567    6774    6789    6854    6869    6934    6943    7009    7018    7070    7085
                         7138    7147    7200    7222    7315    7334    8035#
CLDISK   040522          1459    1503    1584    1708    1901    2051    2302    2369    2466    2523    2584    2644    2678
                         2698    2755    2789    283?    2867    2927    2944    3025    3102    3159    3238    3514    3637
                         3756    3887    3911    4030    4161    4291    4355    4378    4582    4614    4725    4845    4969
                         5092    5219    5345    5467    5558    5918    6001    6073    6183    6263    6437    6571    6652
                         6728    6738    6818    6898    6973    7176    7226    7314    7338    7442    7464    7551    7570
                         7661    7681    7767    8071#   8259    8358    8536    8724    9148
CLR    = 000040            746#    2070    2084    3587    3696    3827    3970    4100    4232    5944    6659    8076
COMMD    045026          5111    5238    5364    5481    5574    6090    6201    6456    6583    7240    7360    8276    8375
                         8557    9223#
COMPA    046064          9363#   9406    9478#   9625#
COMPAR   041416          1952    2337    2432    3071    3139    3207    3567    3617    3679    3727    3810    3854    3952
                         3997    4082    4126    4214    4258    4330    4561    4689    4804    4927    5153    5280    5406
                         5748    5873    6364    6498    6712    7392    7534    7644    7750    8295#   8578    8604
COMMMD   050134          4653    4766    4888    9725#
COUNTD   046532          9619#   9623#   9640
CPHALT   057176          3495    4395    4644    4674    4757    4784    4879    4908    4998    5105    5136    5232    5263
                         5358    5389    5476    5570    5920    6085    6196    6450    6481    6578    6654    7235    7347
                         7490    7596    7702    8271    8371    8538   11056#
CRC      041730          4625    4735    4856    4980    5086    5213    5339    5456    5552    6067    6177    6431    6561
                         7215    7328    8253    8352    8408#   8511
CSF    = 000002            882#
CSU    = 000010            884#
CS1      001702          1022#   2479    2491    2998    5013    8092    8096    8102    8144    8151   11507   11510   11518
                        11525   11531   11541   11544   11548
CS2      001700          1019#   2471    2500    2527    2542    2594    2653    2682    2702    5705    5836    5838    5972
                         6316    6318   11507   11510   11518   11531
CYL      045152          5078#   5087    5205#   5214    5331#   5340    5448#   5457    5545#   5553    6060#   6068    6170#
                         6178    6423#   6432    6554#   7207#   7216    7320#   7329    8246#   8254    8344#   8353    8505#
                         8512    9251#
DAREAD   045226          9291    9302#
DATENV   043234          8801#   9506#   9523#   9573#   9588#  11560
DAWORD   045232          5084#   5211#   5337#   5550#   6065#   6175#   6429#   8251#   8349#   8510#   9303#
DB       001672          1016#  11518
DCK    = 100000            817#   6517    8633    8638    8654    8657
DCL    = 000100            944#
DCLEAR   002032          1090#   3275
DENVL  = 000200            827#
DE1    = 000040            789#
DFF20  = 000002            785#
DF1      064556            305     424     431   11568#
DF11     064574            413     712   11575#
DF14     064603            444   11578#
DF15     064613            453   11581#
DF16     064616            464   11582#
```

DZRPSD.P11 CROSS REFERENCE TABLE

```
DF17    064623          477   11584#
DF2     064563        11570#
DF20    064632          490   11588#
DF21    064641          501   11591#
DF22    064646          516   11593#
DF24    064652          537     550   11595#
DF26    064661          563   11598#
DF27    064671          578   11601#
DF3     064567          393     400   11572#
DF30    064677          591   11604#
DF31    064705          356     364   11606#
DF32    064715          340     381   11609#
DF33    064726          323   11612#
DF34    064736          609   11615#
DF35    064745          628     700   11618#
DF36    064755          643     660   11621#
DF37    064763          683   11623#
DF40    064774          721   11627#
DF5   = 000001          784#
DH1     057461          300     419   11091#
DH11    057727          405     704   11120#
DH14    060107          436   11140#
DH15    060307          450   11163#
DH16    060340          458   11168#
DH17    060503          469   11186#
DH2     057604          387   11106#
DH20    060666          482   11206#
DH21    061051          495   11226#
DH22    061171          511   11241#
DH24    061274          530     543   11253#
DH26    061457          554   11273#
DH27    061662          570   11296#
DH30    062025          584   11314#
DH31    062170          346   11332#
DH32    062372          329     370   11354#
DH33    062615          314   11380#
DH34    063020          599   11403#
DH35    063203          619     690   11423#
DH36    063406          634     651   11446#
DH37    063545          671   11463#
DH40    063763          717   11489#
DIG8  = 000004          786#
DISK    047070         1912    1920    1954    1965    4691    4806    4929    5063    5190    5316    5442    5493    5534
                       5535    5680    5681    5685    5692    5693    5809    5815    5822    5823    6049    6050    6163
                       6164    6286    6292    6299    6300    6304#   6408    6550    6593    7201    7252    7316    7413
                       8492    8606    9049    9063#   9064#   9646    9707#  10140   11551
DISPLA= 177570            69#  10240#  10536#
DLT   = 100000          756#    2469    2653    2732    2735
DL64  = 000020          788#
DMD   = 000001          821#    1712    1715    1736    2067    2304    2384    2928    2945    3037    3103    3160    3253
                       3274    3517    3589    3639    3698    3758    3829    3889    3906    3907    3913    3972    4032
                       4055    4056    4102    4163    4187    4188    4234    4293    4307    4311    4356    4409    4418
                       4540    4583    5007    6685    6689    6739    6819    6899    6974    7177    7374    7443    7467
                       7509    7517    7573    7620    7627    7684    7726    7733    8688    8718    8987    9231    9367
```

```
                        9372    9486    9499    9502    9516    9519    9560    9570    9585    9733    9881    9884    9892
                        9902   10052   10061
DPR    = 000400          792#    2970    5958    5978    6014    7279    7282    8115    8118    8163
DRY    = 000200          791#    2970    3116    3184    3547    3596    3658    3706    3777    3836    3932    3980    4064
                        4109    4196    4241    5958    5978    6014    6691    7279    7282    8115    8123    8158
DST     001706         1024#   4426#   4497#  11528
DS1     001724         1031#    2968    5956    5976    6012    7277    7283    8113    8118    8123    8158    8163   11507
                       11510   11522   11525   11531   11541   11544   11548
DT      001726         1032#
DTAGAP  050074         9710#
DTE    = 010000          814#
DTSY   = 001000          829#    9388    9451    9454
DT1     064040           304     423     430   11500#
DT11    064102           412     711   11507#
DT14    064122           443   11510#
DT15    064144           452   11513#
DT16    064154           463   11515#
DT17    064172           476   11518#
DT2     064054         11502#
DT20    064212           489   11522#
DT21    064232           500   11525#
DT22    064246           515   11527#
DT24    064254           536     549   11528#
DT26    064274           562   11531#
DT27    064316           577   11534#
DT3     064066           392     399   11504#
DT30    064334           590   11538#
DT31    064352           355     363   11541#
DT32    064374           339     380   11544#
DT33    064420           322   11548#
DT34    064442           607   11551#
DT35    064462           626     698   11554#
DT36    064504           641     658   11557#
DT37    064522           681   11560#
DT40    064546           720   11565#
DVA    = 004000          777#    1596    1599    2403    2479    2491    2761    2763    2770    2772    6207    6209    8007
                        8092    8096    8151
ECDATA  043214         8782#    8830    8988#   9055#   9057#   9496#   9498#   9513#   9515#   9563#   9567#   9578#   9582#
ECH    = 000100          808#
ECI    = 004000          908#    3050    5098    5225    5351    5565    6080    6191    6443    8266    8366    8554
ECORR   043640         8982#
ECTEST  043252         8821#    9016    9058    9507    9524    9574    9589
EC1     001732         1034#   11554   11557   11560
EC2     001734         1035#   11554   11557   11560
EMTVEC= 000030          157#    1142#   1143#   1229#   1258#
EM1     054216           299     568   10781#
EM11    054367           404     417     552   10800#
EM14    054422           435   10805#
EM15    054442           448   10808#
EM16    054513           457   10815#
EM17    054534           468   10818#
EM2     054301           312     327     344   10790#
EM20    054560           481   10822#
EM21    054601           494   10825#
```

DZRPSD.P11 CROSS REFERENCE TABLE

```
EM22    054623          505   10829#
EM24    055272          527   10882#
EM25    055365          540   10892#
EM30    055425          583   10898#
EM31    055547          596   10913#
EM32    055672          613   10928#
EM33    056161          633   10960#
EM34    056233          646   10968#
EM35    056326          665   10978#
EM36    056625          688   11011#
EM37    056736          703   11024#
EM40    057073          715   11042#
EM6     054330          368     386   10794#
ERCLFC  017462         3287    3298    3309    3323    3339    3349    3359    3369    3379    3389    3399    3409    3418
                       3429    3438    3449    3459    3469    3486#
ERCRC   050314         9751#   9799#   9973#
ERCS2C  012560         2096    2106    2118    2129    2139    2149    2159    2169    2179    2189    2198    2208    2219
                       2229    2241    2252    2263    2281#
ERFLGS  001774         1050#   1907#   2052#   2598#   2612    4650#   4671    4685#   4763#   4781    4800#   4885#   4905
                       4923#   5004#   5110#   5133    5237#   5260    5363#   5386    5480#   5490    5502    5573#   5584
                       5607    5746#   5852#   6089#   6100#   6120    6200#   6305#   6343#   6455#   6478    6582#   6590
                       6601    7239#   7249    7261    7351#   7408#   7421    7496#   7602#   7708#   8316    8556#   8567
                       9288    9833   10532#
ERHDGP  050316         9752#   9807#   9984#
ERHEAD  050312         9750#   9787#   9961#
ERPOS   043636         8978#   8984#   9001
ERR     = 040000        798#   2970    5958    5978    6014    6692    7375    7518    7628    7734
ERRVEC= 000004          150#   1231#   1238#   1260#  10202   10203#  10205#  10208#
ERSTAR  045000         9148#
ERUNIT  044776         9147#   9149
ERWORD  045272         2345    2440    3079    3147    3215    3575    3626    3686    3736    3817    3863    3960    4006
                       4090    4136    4222    4267    4338    4569    5496#   5501#   5601#   5606#   6114#   6119#   6595#
                       6600#   6719    7255#   7260#   7400    7415#   7420#   7542    7652    7758    8309#   8315#   9354#
                       9390#   9410#   9414#   9436#   9456#   9911#   9918#   9922#   9944#   9952#   9958#   9972#   9977#
                       9981#   9993#  11502   11504   11541   11544   11548
ER1     001704         1023#   2949    2960    6022    6517    6620   11507   11510   11522   11531   11541   11544   11548
                      11554
ER2     001710         1025#  11522
ER3     001716         1028#  11522
EXT1    = 000001        862#
EXT10   = 000010        865#
EXT2    = 000002        863#
EXT20   = 000020        866#
EXT4    = 000004        864#
EXT40   = 000040        867#
FEN     = 000200        888#
FER     = 000020        806#   6517    6620
FILLEC  044012         9039#
FIRST   002022         1079#   1159    1206#
FMT22   = 010000        909#   3050    4384    4619    4635    4651    4729    4745    4764    4850    4867    4886    4974
                       4990    5005    5078    5098    5139    5205    5225    5266    5331    5351    5392    5448    5473
                       5545    5565    5656    5723    6060    6080    6170    6191    6443    6575    6670    6782    6862
                       7078    7207    7232    7320    7344    7478    7497    7584    7603    7709    8246    8266    8344
                       8366    8505    8554    8682
```

```
FNWORD  050334          4624#  4734#  4855#  4979#  9837#
FORMAT  046534          9620#  9639
FOUT    050340          9834   9840#
FRMAT1  032514          6549#
FSYNER  050310          9749#  9774#  9947#
FUTABL  002024          1096#  2930
GCRC    050304          4627   4737   4858   4982   6563   9760#  9968   9971
GECC1   043216          8786#  8828   8914#  9047#  9063   11551  11554  11560
GECC2   043220          8789#  8829   8915#  9048#  9064   11551  11554  11560
GNS   = ****** U         170   1167   1171   1175   1179   1183   1187   1191   1195   1199   1203   1212   1247
                        1251   1321   1325   1329   1333   1337   1365   1369   1373   1377   1381   1412   1419
                        1425   1466   1474   1481   1508   1512   1516   1522   3786   7782   7788   7889   7895
                        7902   7906   7910   7917   7921   9085   9091   9103   9109   9115   9119   9123   9129
                        9135  10163  10719  10720  10721  10722  10723  10724  10725  10726  10727  10728  10729
GO    = 000001           771#  2314   2958   3046   3113   3181   3276   3541   3545   3594   3652   3656   3704
                        3772   3775   3834   3927   3930   3977   4050   4062   4107   4182   4194   4239   4306
                        4410   5008   5010   5966   6693   7510   7621   7727   8689   8719   9232   9734
GRV   = 000010           787#
HADTMP  043240          8811#  9003
HARDER  043232          8798#
HCCRCE  042332          6799   6879   6953   7028   7095   7157   8534#
HCE   = 000200           809#  8623   8627   8633   8638
HCI   = 002000           907#  3050
HCRC  = 000400           810#  8645   8648   8654   8657
HDESYN  050320          9753#  9820#  9996#
HDMSYN  047066          9705#
HEADER  047042          9702#
HEDGAP  045166          9247#  9266#  9417#
HEDSYN  045170          9248#  9277#  9437#  9458#
HEGAP   047054          9408   9704#
IAE   = 002000           812#  7514   7624   7730
IE    = 000100           772#
ILF   = 000001           802#  2960
ILLEGL  002066         1104#  2941#  2946
ILR   = 000002           803#
IOTVEC= 000020           155#  1140#  1141#
IR    = 000100           747#  1606   2401   2469   2499   2525   2540   2680   2700   2735   2795   2801   2811
                        5703   5834   6215   6314   7367   8012
IXE   = 004000           892#
KEY1    045156          5082#  5209#  5335#  5452#  5548#  6063#  6173#  6427#  6557#  7211#  7324#  8249#  8347#
                        8508#  9253#
KEY2    045160          5083#  5210#  5336#  5453#  5549#  6064#  6174#  6428#  6558#  7212#  7325#  8250#  8348#
                        8509#  9254#
KIPAR0= 172340           210#
KIPAR1= 172342           211#
KIPAR2= 172344           212#
KIPAR3= 172346           213#
KIPAR4= 172350           214#
KIPAR5= 172352           215#
KIPAR6= 172354           216#
KIPAR7= 172356           217#
KIPDR0= 172300           199#
KIPDR1= 172302           200#
KIPDR2= 172304           201#
```

DZRPSD.P11 CROSS REFERENCE TABLE

```
KIPDR3= 172306          202#
KIPDR4= 172310          203#
KIPDR5= 172312          204#
KIPDR6= 172314          205#
KIPDR7= 172316          206#
LA      001736         1036#
LAD     040502         7984*  8053#  8057
LERR    040170         7967#  7973*  8014
LST   = 002000          794#  7279   7282   7375
MAKECY  043074         4357   4584   6740   6820   6900   6975   7178   7444   8710#
MASK    040166         7966#  7970*  7994*  7995   7996*  7999*  8000   8001*
MCLK  = 000002          822#  4449   4450   4451   4452   4478   4479   4480   4481   4527   4528   4531   4532
                       9014   9015   9022   9023   9369   9370   9372   9373   9375   9376   9491   9508   9560
                       9575   9884   9894  10079  10081  10082  10083  10084  10098  10099  10100  10101
MCPE  = 020000          778#
MHS   = 001000          890#
MID     043064         8684*  8691#
MIDDLE  042762         2985   8673#
MINX  = 000004          823#  3894   3895   4037   4038   4170   4171   4417   7468   7469   7574   7575   7685
                       7686  10060
MMVEC = 000250          188#
MOL   = 010000          796#  1504   1518
MPE   = 000400          749#
MR      001722         1030#  11557
MRD   = 000020          825#  9489   9494   9502   9511   9519
MSE   = 000020          885#
MSTCK = 000010          824#  3906   4055   4187   4411   4412   4414   4415   4417   4449   4450   4478   4479
                       4513   4514   4516   4517   4531   4532   9368   9370   9372   9373   9491   9560   9883
                       3884  10053  10054  10055  10056  10060  10078  10081  10082  10098  10099  10108  10109
MWR   = 000040          826#  9561   9576   9885   9895
MXF   = 001000          750#  2995
NCODE   043224         8795#
NCOUNT  043226         8796#  8994*
NED   = 010000          753#  1922   9151
NEM   = 004000          752#  6215
NHS   = 002000          891#
NOPERA  002024         1087#  3104   3172
NOSYNC  045266         6547#  6766*  6846*  6926*  7001*  7061*  7129*  9292   9351#  9425   9449   10186*
NOUNIT  001764         1043#  1395*  1432*  1439   7802*
NOWORD  045220         5454#  6559*  7213*  7326*  9299#
NUNIT   001766         1045#  1439*  1440*  3330
OCYL  = 100000          946#
OF      001712         1026#
OFREV = 000200          906#
OFSETC  002056         1100#  3917
OF100 = 000004          901#
OF200 = 000010          902#
OF25  = 000001          899#  3918   3978
OF400 = 000020          903#
OF50  = 000002          900#
OF800 = 000040          904#
OPERSE  037346         7885#  10396
OPI   = 020000          815#
OR    = 000200          748#  2538   2540   2592   2697   2713   2735   6314   7366   8012
```

```
OUT     045236          9289    9296    9301    9305#
PAR   = 000010           805#   1616    1619
PAT   = 000020           745#   1588    1606
PC    =X000007            81#   1158#   1283#   1315#   1459#   1503#   1544#   1569#   1584#   1645#   1669#   1693#   1708#
                        1775#   1801#   1826#   1851#   1872#   1901#   2051#   2096#   2106#   2118#   2129#   2139#   2149#
                        2159#   2169#   2179#   2189#   2198#   2208#   2219#   2229#   2241#   2252#   2263#   2286#   2302#
                        2352#   2369#   2395#   2447#   2466#   2470#   2490#   2523#   2526#   2541#   2584#   2593#   2644#
                        2651#   2678#   2681#   2698#   2701#   2755#   2789#   2834#   2867#   2927#   2944#   2948#   2959#
                        2985#   2996#   3025#   3084#   3102#   3152#   3159#   3220#   3238#   3287#   3298#   3309#   3323#
                        3339#   3349#   3359#   3369#   3379#   3389#   3399#   3409#   3418#   3429#   3438#   3449#   3459#
                        3469#   3491#   3494#   3514#   3582#   3633#   3637#   3692#   3741#   3756#   3824#   3869#   3887#
                        3911#   3966#   4013#   4030#   4096#   4143#   4161#   4228#   4274#   4291#   4344#   4355#   4378#
                        4394#   4573#   4582#   4614#   4643#   4653#   4673#   4697#   4703#   4725#   4756#   4766#   4783#
                        4812#   4822#   4845#   4878#   4888#   4907#   4935#   4945#   4969#   4997#   5012#   5092#   5104#
                        5111#   5135#   5161#   5165#   5219#   5231#   5238#   5262#   5288#   5292#   5345#   5357#   5364#
                        5388#   5414#   5418#   5467#   5475#   5481#   5489#   5558#   5569#   5574#   5696#   5704#   5756#
                        5762#   5826#   5835#   5883#   5889#   5918#   5919#   5947#   5967#   6001#   6005#   6073#   6084#
                        6090#   6183#   6195#   6201#   6263#   6306#   6315#   6374#   6380#   6437#   6449#   6456#   6480#
                        6506#   6510#   6515#   6571#   6577#   6583#   6589#   6652#   6653#   6723#   6728#   6738#   6770#
                        6818#   6850#   6898#   6930#   6973#   7005#   7065#   7133#   7176#   7226#   7234#   7240#   7248#
                        7314#   7338#   7346#   7360#   7405#   7411#   7442#   7464#   7489#   7547#   7551#   7570#   7595#
                        7657#   7661#   7681#   7701#   7763#   7767#   7833#   7836#   7850#   7954#   7976#   7977#   7991#
                        8014#   8016#   8079#   8091#   8132#   8135#   8143#   8168#   8171#   8259#   8270#   8276#   8279#
                        8318#   8320#   8358#   8370#   8375#   8378#   8517#   8536#   8537#   8557#   8566#   8586#   8594#
                        8612#   8617#   8694#   8724#   8935#   8941#   8968#   9009#   9016#   9058#   9071#   9093#   9148#
                        9312#   9383#   9387#   9395#   9397#   9399#   9401#   9403#   9411#   9422#   9444#   9448#   9507#
                        9524#   9530#   9574#   9589#   9601#   9647#   9653#   9659#   9663#   9666#   9848#   9919#   9939#
                        9953#   9966#   9978#   9989#  10142#  10150#  10346#  10353#  10358#  10385#  10395#  10546#  10613#
                       10763
PCJSR   002002          1059#   8089#   8090#   8141#   8142#   8534#   8535#   8712#   8713#   8982#   8983#   9223#   9224#
                        9725#   9726#  11531   11534   11538   11544   11548   11557
PGE   = 002000           751#   5013
PIE1  = 100000          8749#   8898    8913
PIE10 = 000100          8758#
PIE11 = 000040          8759#   8865
PIE12 = 000020          8760#   8898
PIE13 = 000010          8761#
PIE14 = 000004          8762#
PIE15 = 000002          8763#
PIE16 = 000001          8764#
PIE17 = 100000          8765#
PIE18 = 040000          8766#
PIE19 = 020000          8767#
PIE2  = 040000          8750#   8856
PIE20 = 010000          8768#
PIE21 = 004000          8769#   8873
PIE22 = 002000          8770#   8904
PIE23 = 001000          8771#   8881
PIE24 = 000400          8772#   8904
PIE25 = 000200          8773#
PIE26 = 000100          8774#
PIE27 = 000040          8775#
PIE28 = 000020          8776#
PIE29 = 000010          8777#
```

```
PIE3   = 020000          8751#  8898
PIE30  = 000004          8778#
PIE31  = 000002          8779#
PIE32  = 000001          8780#  8837  8845
PIE4   = 010000          8752#
PIE5   = 004000          8753#
PIE6   = 002000          8754#
PIE7   = 001000          8755#
PIE8   = 000400          8756#
PIE9   = 000200          8757#
PIP    = 020000           797#  3546  3595  3657  3705  3776  3835  3931  3979  4063  4108  4195  4240
PIRQ   = 177772            67#
PIRQVE= 000240            161#
PKACK    002062          1102#  2305
PLU    = 020000           894#
POSITI   043230          8797#  8938  8992  9000#  9001  9003  9046#  11554  11560
PRBA     010440          1663#  1664#
PRCA     011112          1845#  1846#
PRCS1    010160          1563#  1564#
PRCS2    010114          1277   1538#  1539#
PROST    010736          1769#  1770#
PRE    = 000020           942#
PRER1    010504          1687#  1688#
PRER2    011002          1795#  1796#
PRER3    011156          1866#  1867#
PROF     011046          1820#  1821#
PROG   = 001000           793#  2216  2413  2969  3426  5957  5977  6013  7278  8114
PRWC     010374          1639#  1640#
PR0    = 000000            84#
PR1    = 000040            85#
PR2    = 000100            86#
PR3    = 000140            87#
PR4    = 000200            88#
PR5    = 000240            89#
PR6    = 000300            90#
PR7    = 000340            91#
PS     = 177776            64#    65  1134#  1156#  7779#  7886#  10419#
PSEL   = 002000           776#
PSU    = 000001           939#
PSW    = 177776            65#
PUTREG   040126          2470  2490  2526  2541  2593  2651  2681  2701  2948  2959  2996  5012  5489
                         5704  5835  5947  5967  6005  6315  6515  6589  7248  7411  7941#  8091  8143
                         8566  8935  8941  9009
PWRVEC= 000024             156#  1146#  1147#  10735#  10736#  10744#  10759#  10760#
P12      043244          8813#  8867#  8868#  8897
P22      043246          8814#  8875#  8876#  8902
P24      043250          8815#  8883#  8884#  8903
P3       043242          8812#  8858#  8859#  8896
RCOM     051456         10128#  10133#
RCYL     045256          9321#  9359#  9394
RDCHR  = 104412         10440   10724#
RDHEAD   045274          9249   9359#
RDLIN  = 104414         10482   10725#
RDOCT  = 104416          1214   7912  7923  9094  9111  10726#
```

DZRPSD.P11 CROSS REFERENCE TABLE

```
RDY    = 000200   773#  1596  1599  2403  2479  2491  2761  2763  2770  2772  3120  3188  6207
                 6209  6694  7292  8007  8092  8102  8144
READ     046066  9383  9387  9395  9397  9399  9401  9403  9444  9448  9483#  10142  10150
READAT   002050  1097#  5559  6074  6184
READIN   002064  1103#  3035
RECALI   002030  1089#  4173
REDATA   051460  9302  10132#
REFOR    002052  1098#  5109  5236  5362  6454  7494
REGADR   040172  1303#  1600#  1611#  1620#  1720#  1742#  2077#  2281#  2347#  2442#  2555#  2562#  2611#
                 2661#  2717#  2725#  2738#  2764#  2773#  2802#  2817#  2844#  2852#  2877#  2885#  3081#
                 3149#  3217#  3486#  3577#  3628#  3688#  3738#  3819#  3865#  3962#  4008#  4092#  4138#
                 4224#  4269#  4340#  4571#  5970#  6210#  6220#  6228#  6721#  7281#  7402#  7544#  7654#
                 7760#  7968#  7972#  8626#  8637#  8647#  8658#  8729#  11500  11534  11538
REGSA1   053246  1229  10533#
REINTO   003134  1110#  2310  2317#  2331  2338  2390  2399#  2402#  2403#  2404#  2405#  2406#  2407#
                 2408#  2409#  2412  2416#  2417#  2418#  2429  2433  3042  3049#  3050#  3052#  3053#
                 3066  3072  3537  3545#  3546#  3547#  3562  3568  3594#  3595#  3596#  3597#  3598#
                 3613  3618  3648  3656#  3657#  3658#  3673  3680  3704#  3705#  3706#  3707#  3708#
                 3722  3728  3768  3775#  3776#  3777#  3804  3811  3834#  3835#  3836#  3837#  3851
                 3855  3923  3930#  3931#  3932#  3946  3953  3977#  3978#  3979#  3980#  3981#  3994
                 3998  4046  4062#  4063#  4064#  4077  4083  4107#  4108#  4109#  4110#  4122  4127
                 4178  4194#  4195#  4196#  4209  4215  4239#  4240#  4241#  4242#  4254  4259  4302
                 4311#  4312#  4325  4331  4391  4537#  4538#  4540#  4541#  4542#  4556  4562  4682
                 4683  4690  4792  4793  4796  4797  4805  4916  4917  4920  4921  4928  5094
                 5155  5221  5282  5347  5408  5539  5540  5561  5599  5655  5661  5662  5666
                 5673  5674  5750  5790  5795  5802  5803  5875  5926  5940  6054  6055  6076
                 6112  6267  6272  6279  6280  6335  6366  6439  6500  6677  6689#  6690#  6691#
                 6692#  6693#  6694#  6707  6713  6775  6776  6805  6855  6856  6885  6944  6945
                 7019  7020  7071  7072  7101  7148  7149  7356  7363#  7364#  7366#  7367#  7368#
                 7369#  7370#  7372#  7373#  7374#  7375#  7387  7393  7475  7503  7513#  7514#  7515#
                 7516#  7517#  7518#  7529  7535  7614  7624#  7625#  7626#  7627#  7628#  7639  7645
                 7720  7730#  7731#  7732#  7733#  7734#  7745  7751  8262  8362  8580  8605
RELEAS   002034  1091#  4297
RESVEC=  000010  151#
RETCL    002060  1101#  4041
RHAS     001650  1000#  1297  1340  5935  6004#
RHBA     001626  983#  1663  2058#  2375#  2868  3030#  3244#  4633#  4743#  4865#  4988#  5094#  5221#
                 5347#  5469#  5561#  6076#  6186#  6223  6228  6439#  6573#  6667#  7228#  7340#  7476#
                 7582#  7693#  8262#  8362#  8551#  8679#
RHCA     001644  998#  1845  2065#  2382#  3033#  3251#  3531#  3644#  3900#  4383#  4652#  4765#  4887#
                 5006#  5102#  5229#  5355#  5474#  5568#  6083#  6194#  6447#  6576#  6657  6671#  7233#
                 7345#  7498#  7604#  7710#  8269#  8369#  8543#  8683#  8715#
RHCC     001670  1008#  3525  3900  7947  8725  8729
RHCS1    001632  993#  1563  1930  2060#  2305#  2314#  2377#  3036#  3046#  3246#  3524#  3541#  3643#
                 3652#  3762#  3772#  3917#  3927#  4041#  4050#  4173#  4182#  4297#  4306#  4410#  5008#
                 5010#  5966#  6664#  6676  6686#  6700  6721  7289  7510#  7621#  7727#  8005  8071
                 8676#  8689#  8717#  8719#  9087  9125  9232#  9734#
RHCS2    001630  984#  1278  1341  1401#  1538  1539#  1564#  1588#  1640#  1664#  1688#  1770#  1796#
                 1821#  1846#  1867#  2059#  2376#  2396#  2535  2694  3168#  3245#  5015  5944#  5945#
                 6088#  6199#  6659#  6660#  6669#  7990#  8009  8072  8681#
RHDB     001622  981#  1233  2056#  2072  2467  2533#  2534#  2548  2549  2555  2562  2587#  2601
                 2611  2647#  2657  2661  2687  2707  3242#  3277  9095
RHDST    001636  995#  1769  2062#  2379#  3031#  3248#  3523#  4381#  4406  4426  4497  4640#  4754#
                 4876#  4995#  5097#  5224#  5350#  5472#  5564#  6079#  6190#  6442#  6574#  6668#  7231#
```

DZRPSD.P11 CROSS REFERENCE TABLE

```
                        7343*  7370   7487*  7593*  7699*  8265*  8365*  8548*  8680*  8716*
RHDS1   001654          1002*  2977   3111   3179   7281   8073
RHDT    001656          1003*  1402   1404   1421   1483   1487   1489   1937   1939
RHEC1   001662          1005*  8938
RHEC2   001664          1006*  3107   3166   3175   8932
RHER1   001634           994*  1344   1590*  1687   2061*  2378*  3247*  5928   7613   7632   7654   7719   7738
                        7760   8074   8622   8626   8632   8637   8644   8647   8652   8658
RHER2   001640           996*  1795   2063*  2380*  3249*  5930
RHER3   001646           999*  1866   2066*  2383*  3252*  5932
RHLA    001666          1007*  3597   3707   3837   3981   4110   4242   4312   4425   4495
RHMR    001652          1001*  1709   2067*  2068*  2304*  2384*  2385*  2928*  2945*  3037*  3103*  3160*  3253*
                        3254*  3274*  3517*  3589*  3639*  3698*  3758*  3829*  3889*  3894*  3895*  3906*  3907*
                        3913*  3972*  4032*  4037*  4038*  4055*  4056*  4102*  4163*  4170*  4171*  4187*  4188*
                        4234*  4293*  4307*  4356*  4405   4583*  5007*  6685*  6739*  6819*  6899*  6974*  7177*
                        7443*  7467*  7468*  7469*  7508   7573*  7574*  7575*  7619   7684*  7685*  7686*  7725
                        8688*  8718*  8986   9231*  9365   9638   9733*  9880   10050
RHOF    001642           997*  1820   2064*  2381*  3032*  3250*  3918*  4384*  4651*  4764*  4886*  5005*  5098*
                        5225*  5351*  5473*  5565*  6080*  6191*  6443*  6575*  6670*  7232*  7344*  7497*  7603*
                        7709*  8266*  8366*  8554*  8682*
RHSN    001660          1004*  1476   1486   1488
RHWC    001624           982*  1639   2057*  2309   2324   2347   2374*  2389   2422   2442   2835   3029*  3041
                        3059   3081   3105   3107   3126   3149   3165   3166   3173   3175   3194   3217   3243*
                        3536   3555   3577   3605   3628   3647   3665   3688   3715   3738   3767   3797   3819
                        3843   3865   3922   3939   3962   3987   4008   4045   4070   4092   4115   4138   4177
                        4202   4224   4247   4269   4301   4318   4340   4390   4549   4571   4631*  4666   4741*
                        4776   4862*  4898   4986*  5093*  5123   5220*  5250   5346*  5376   5468*  5560*  6075*
                        6185*  6438*  6468   6572*  6666*  7227*  7339*  7355   7380   7402   7474*  7502   7522
                        7544   7580*  7691*  7945   7947   8261*  8361*  8549*  8678*
RKEY1   045262          9323*  9361*  9398
RKEY2   045264          9324*  9362*  9400
RMR     = 000004         804*  6690
RNCTR1  050132          9724*  9736*  9738*
RNO     051454         10127*  10132*
RNWAT1  050176          9736*
RPVEC   001620           971*  1157*  9105   9112*  9131   10166*
RPVECT  051570          1157   10160*  10166
RSETR   045260          9322*  9360*  9396
RSYNC   045254          9320*  9386   9391   9423   9434   9447   9457   9942   9945   9991   9995
RUNCTR  045024          9222*  9235*  9237*
RUNWAT  045070          9235*
R0      =%000000          72*  1232*  1235*  1343*  1347*  1387*  1390*  1396*  1435*  1461*  1462   1709*  1712*
                        1713*  1717   1719   1720   1736*  1737*  1738   1741   1742   1913*  1915*  1919*  1924*
                        1944   1947*  1952*  2072*  2077   2078   2093   2103   2115   2126   2136   2146   2156
                        2166   2176   2186   2194   2205   2214   2226   2238   2249   2260   2269   2274   2281
                        2285   2308*  2323*  2337*  2388*  2421*  2432*  2467*  2548*  2550   2554   2579*  2581*
                        2585*  2587   2588*  2599*  2601*  2604*  2607   2609   2623   2645*  2646*  2647   2648
                        2687*  2689*  2707*  2708*  2710   2711   2717   2725   2930*  2932   3040*  3058*  3071*
                        3105*  3108   3125*  3139*  3165*  3167*  3173*  3176   3193*  3207*  3259*  3262*  3277*
                        3284   3295   3306   3320   3335   3346   3356   3366   3376   3386   3396   3406   3415
                        3424   3435   3446   3456   3466   3475   3480   3486   3490   3493   3535*  3554*  3567*
                        3604*  3617*  3646*  3664*  3679*  3714*  3727*  3766*  3796*  3810*  3842*  3854*  3905*
                        3908*  3921*  3938*  3952*  3986*  3997*  4044*  4054*  4057*  4069*  4082*  4114*  4126*
                        4176*  4186*  4189*  4201*  4214*  4246*  4258*  4300*  4317*  4330*  4357*  4388*  4405*
                        4409*  4411*  4412*  4414*  4415*  4417*  4418*  4449*  4450*  4451*  4452*  4478*  4479*
```

```
                4480* 4481* 4513* 4514* 4516* 4517* 4527* 4528* 4531* 4532* 4548* 4561* 4584*
                4609* 4611* 4632* 4633  4635* 4637* 4681* 4689* 4720* 4722* 4742* 4743  4745*
                4747* 4748* 4749* 4751* 4791* 4795* 4804* 4840* 4842* 4863* 4865  4867* 4869*
                4870* 4871* 4873* 4915* 4919* 4927* 4964* 4966* 4987* 4988  4990* 4992* 5063*
                5064* 5071* 5138* 5139* 5142* 5143* 5144* 5147* 5153* 5190* 5191* 5198* 5265*
                5266* 5269* 5270* 5271* 5274* 5280* 5316* 5317* 5324* 5391* 5392* 5395* 5396*
                5397* 5400* 5406* 5441* 5463* 5492* 5497  5499  5533* 5538* 5586* 5591* 5598*
                5602  5604  5622  5660* 5665* 5667  5669* 5672* 5679* 5684* 5686  5688* 5691*
                5722* 5723* 5724* 5725* 5726* 5728* 5733* 5735  5737* 5740* 5748* 5789* 5791
                5792* 5794* 5796  5798* 5801* 5808* 5810  5811* 5814* 5816  5818* 5821* 5853*
                5855  5856* 5859* 5861  5863* 5866* 5873* 5926* 5928* 5929* 5930* 5931* 5932*
                5933* 5940* 5942  5989  5993  6048* 6053* 6101* 6111* 6115  6117  6135  6162*
                6266* 6268  6269* 6271* 6273  6275* 6278* 6285* 6287  6288* 6291* 6293  6295*
                6298* 6344* 6346  6347* 6350* 6352  6354* 6357* 6364* 6408* 6409* 6416* 6483*
                6484* 6487* 6488* 6489* 6492* 6498* 6549* 6567* 6592* 6596  6598  6657* 6658
                6675* 6699* 6712* 6740* 6774* 6781* 6782* 6783* 6784* 6785* 6789* 6799* 6820*
                6854* 6861* 6862* 6863* 6864* 6865* 6869* 6879* 6900* 6934* 6943* 6953* 6975*
                7009* 7018* 7028* 7070* 7077* 7078* 7079* 7080* 7081* 7085* 7095* 7138* 7147*
                7157* 7178* 7200* 7222* 7251* 7256  7258  7315* 7334* 7354* 7379* 7392* 7412*
                7416  7418  7444* 7475* 7476  7478* 7480* 7481* 7482* 7484* 7501* 7508* 7509*
                7521* 7534* 7581* 7582  7584* 7586* 7587* 7588* 7590* 7612* 7619* 7620* 7631*
                7644* 7692* 7693  7696* 7718* 7725* 7726* 7737* 7750* 7804* 7806  7843* 7845
                7847  7850  7942  7945* 7948  7953* 8039  8040  8041  8051* 8180  8181* 8182
                8184  8185  8186  8187* 8236  8227  8228  8235* 8301  8302  8303  8304  8305
                8306* 8334* 8409  8410* 8418  8470* 8487  8490* 8493  8516* 8534  8541  8542
                8543  8544  8545  8546  8547  8549  8551  8552  8578* 8604* 8618  8662* 8674
                8693* 8712  8714  8734* 8822  8839* 8846* 8849* 8850* 8851  8896* 8897* 8899
                8901* 8902* 8903* 8905  8967* 8982  8984  9027* 9040  9070* 9095* 9097* 9098*
                9225  9311* 9365* 9367* 9368* 9369* 9370* 9372* 9373* 9375* 9376* 9388  9451
                9454  9486* 9489* 9491* 9494  9503* 9508* 9511  9520* 9552  9557* 9560* 9561
                9570* 9575* 9576  9585* 9600* 9627  9638* 9675* 9727  9846* 9880* 9881* 9883*
                9884* 9885  9892* 9894* 9895  9902* 10044 10050* 10052* 10053* 10054* 10055* 10056*
                10060* 10061* 10078* 10079* 10081* 10082* 10083* 10084* 10098* 10099* 10100* 10101* 10108*
                10109* 10119* 10258 10268* 10272 10288 10289 10302* 10338 10339* 10340 10343* 10479
                10483* 10484 10487 10507* 10510* 10569 10570* 10571* 10578* 10579* 10580* 10581* 10582*
                10583  10588  10594  10596* 10597  10611* 10703  10704* 10705  10706* 10707* 10708* 10737
                10758*
R1      =%000001  73*  1233* 1234  1240  1297* 1298* 1299  1303  1340* 1350  1353  1397  1595
                1600  1710* 1713  1714  1725* 1726* 1727* 1734* 1737  1738  1740  1747* 1748*
                1749* 1912* 1914* 1920* 1945* 1964* 1966  2482  2488* 2494  2688* 2689  2691*
                2692  2709* 2713  2756* 2757  2760  2764  2768* 2769  2773  2946* 2958* 3001
                3104* 3112  3113* 3172* 3180  3181* 3275* 3276* 3901* 3902* 4385* 4475* 4486*
                4509* 4610* 4612* 4648* 4721* 4723* 4761* 4841* 4843* 4883* 4965* 4967* 5002*
                5109* 5145* 5148* 5236* 5272* 5275* 5362* 5398* 5401* 5479* 5493* 5497  5500
                5508  5559* 5599* 5602  5605  5622  5655* 5656* 5657* 5658* 5659* 5666* 5667*
                5685* 5686* 5734* 5735* 5790* 5791* 5795* 5796* 5809* 5810* 5815* 5816* 5854*
                5855* 5860* 5861* 5943* 5946  5968  5971  5987* 5989  6074* 6112* 6115  6118
                6135  6184* 6187* 6205  6210  6267* 6268* 6272* 6273* 6286* 6287* 6292* 6293*
                6345* 6346* 6351* 6352* 6454* 6490* 6493* 6581* 6593* 6596  6599  6610  7238*
                7252* 7256  7259  7267  7350* 7413* 7416  7419  7426  7494* 7600* 7706* 7805*
                7806  7809  7943  7946* 7948* 7952* 8036  8039* 8042  8044* 8050* 8071* 8078*
                8098  8104  8108  8147  8154  8223  8226* 8229  8234* 8274* 8296  8301* 8310
                8313  8321  8333* 8374* 8411  8415* 8423  8430  8437  8445* 8448* 8450* 8453*
                8455* 8458* 8460* 8465  8469* 8488  8491* 8494* 8496* 8500* 8515* 8542* 8675
```

```
                    8590*  8692*  8823   8828*  8836   8844   8872   8880   8895*  8904*  8905*  8912*  8914
                    8930   8966*  8985   8986*  8987*  9006   9014*  9015*  9022*  9023*  9026*  9041   9049*
                    9052   9069*  9096*  9099*  9226   9243*  9215*  9246   9247   9248   9249*  9297*  9302*
                    9310*  9359   9360   9361   9362   9363   9364   9460*  9461*  9623   9624   9625   9628
                    9633*  9636*  9658*  9661*  9665*  9668*  9674*  9677*  9728   9744*  9747*  9748   9749
                    9750   9751   9752   9753   9755*  9836*  9845*  9873   9874   9875   9876   9877   9878
                    9879*  9908*  9920*  9941*  9942   9946   9954*  9955   9960   9967*  9968   9970   9979*
                    9983   9990*  9991   9994   9999*  10001*  10043  10045  10095*  10106*  10118*  10120*  10132
                   10133  10134  10154*  10155*  10259  10272*  10273  10277  10301*  10480  10485*  10493*  10495*
                   10497*  10500*  10503  10506*  10593  10594*  10599  10603  10605  10610*  10738  10757*
R2      =%000002     74*  1234*  1341*  1342*  1346*  1608   1611   1711*  1728*  1735*  1750*  1918*  1922
                    1927   1934*  1949*  1965*  1968   2070*  2071*  2084*  2087*  2473   2502   2529   2544
                    2596   2655   2684   2704   2709   2712   2738   2790*  2791   2796   2802   2807*  2809*
                    2813   2817   2992   3107*  3109*  3175*  3177*  3260*  3265*  3271*  3587*  3588*  3696*
                    3697*  3827*  3828*  3970*  3971*  4100*  4101*  4232*  4233*  4446*  4453*  4456*  4477*
                    4482*  4526*  4529*  5146*  5147   5273*  5274   5399*  5400   5494*  5501   5513*  5600*
                    5606   5627*  5707   5942*  5946*  5968   5970   6113*  6119   6140*  6217   6220   6491*
                    6492   6594*  6600   6615*  7253*  7260   7272*  7414*  7420   7432*  7944   7947*  7949*
                    7951*  8037   8040*  8042*  8043*  8045*  8046*  8049*  8072*  8076*  8077*  8224   8227*
                    8229*  8233*  8297   8302*  8310   8314   8321   8332*  8412   8432*  8434*  8435*  8451
                    8468*  8489   8492*  8493*  8498*  8514*  8824   8829*  8855   8864   8894*  8898*  8899*
                    8911*  8913*  8915   8965*  9042   9050*  9061*  9068*  9149*  9151   9227   9309*  9374*
                    9377*  9381*  9384*  9409*  9414   9420*  9442*  9445*  9484   9490*  9525*  9529*  9553
                    9559*  9590*  9599*  9624*  9629   9649*  9673*  9729   9844*  9893*  9903*  9917*  9922
                    9928*  9950*  9958   9964*  9976*  9981   9987*  10046  10076*  10085*  10087*  10097*  10102*
                   10117*  10138*  10143*  10148*  10151*  10260  10271*  10275*  10278  10295*  10286*  10287  10292*
                   10300*  10481  10486*  10494*  10496*  10498*  10504  10505*  10739  10756*
R3      =%000003     75*  1388*  1389*  1393*  1428*  1504   1518   1930*  1931*  1932*  1963*  1996*  2711*
                    2722   2727   3106*  3108*  3174*  3176*  4406*  4407*  4408   4419   4431   4501   4519*
                    7975*  7979*  7980   7983*  7993   8038   8041*  8044   8048*  8073*  8120   8125   8128
                    8160   8165   8225   8228*  8230*  8232*  8298   8303*  8307   8315   8326*  8331*  8413
                    8425*  8427*  8428*  8435   8442   8446   8467*  8825   8836*  8837*  8838*  8839   8844*
                    8845*  8846   8855*  8856*  8857*  8858   8864*  8865*  8866*  8867   8872*  8873*  8874*
                    8875   8880*  8881*  8882*  8883   8964*  9043   9051*  9059*  9067*  9228   9308*  9554
                    9565*  9569*  9580*  9584*  9595   9598*  9630   9634*  9635*  9646*  9648*  9654*  9660*
                    9664*  9667*  9672*  9730   9843*  9888*  9891*  9898*  9901*  9908   9909   9913   9951*
                    9955   9959   10047  10051*  10062  10110*  10116*  10140*  10141  10149  10261  10269*  10270*
                   10284*  10287*  10296*  10297*  10299*  10436  10437*  10438  10441*  10442  10446  10448  10450*
                   10452*  10650  10659*  10665*  10666*  10669*  10674*  10675*  10676  10685*  10740  10755*
R4      =%000004     76*  1344*  1345*  1615   1620   1921   2712*  2732   2737   2835*  2836*  2837   2840
                    2844   2847*  2848   2852   2868*  2869*  2870   2873   2877   2880*  2881   2885   2951
                    2962   3264*  5935*  6003*  7971*  7972   7997*  7998   8004   8005   8009   8011   8074*
                    8299   8307*  8308*  8309   8330*  8414   8439*  8441*  8442*  8456   8466*  8826   8963*
                    9044   9052*  9053*  9066*  9150   9229   9307*  9631   9642*  9651   9655*  9671*  9731
                    9842*  10048  10115*  10651  10653*  10654*  10655*  10656  10657*  10671  10673*  10681*  10684*
                   10741  10754*
R5      =%000005     77*  1279*  1394*  1401   1415   1428   1430   1434*  1540*  1565*  1641*  1665*  1689*
                    1771*  1797*  1822*  1847*  1868*  2345*  2346*  2347   2440*  2441*  2442   2549*  2558
                    2561   2580*  2582*  2586*  2589*  2600*  2602*  2605*  2630*  2710*  2715   2719   2931*
                    2936*  3079*  3080*  3081   3147*  3148*  3149   3166*  3170*  3215*  3216*  3217   3261*
                    3266*  3575*  3576*  3577   3626*  3627*  3628   3686*  3687*  3688   3736*  3737*  3738
                    3817*  3818*  3819   3863*  3864*  3865   3960*  3961*  3962   4006*  4007*  4008   4090*
                    4091*  4092   4136*  4137*  4138   4222*  4223*  4224   4267*  4268*  4269   4338*  4339*
                    4340   4447*  4457*  4473*  4508*  4509   4569*  4570*  4571   4625*  4634*  4638*  4735*
```

DZRPSD.P11 CROSS REFERENCE TABLE


```
                          4750*  4752*  4856*  4872*  4874*  4980*  4989*  4993*  5062*  5065*  5068*  5072*  5086*
                          5189*  5192*  5195*  5199*  5213*  5315*  5318*  5321*  5325*  5339*  5456*  5552*  5936*
                          5948   6067*  6177*  6407*  6410*  6413*  6417*  6431*  6561*  6658*  6687*  6719*  6720*
                          6721   7215*  7328*  7400*  7401*  7402   7483*  7485*  7542*  7543*  7544   7589*  7591*
                          7652*  7653*  7654   7695*  7697*  7758*  7759*  7760   7970   7971   7973   7974*  7982*
                          8253*  8300   8329*  8352*  8410   8465*  8471*  8511*  8711   8714*  8715   8726   8728
                          8733*  8827   8962*  9045   9065*  9230   9306*  9366*  9379*  9408*  9419*  9439*  9485*
                          9527*  9555   9556*  9592*  9597*  9732   9841*  9882*  9905*  10049  10077*  10088*  10096*
                          10104*  10114*  10262  10264*  10266*  10273*  10277*  10292  10298*  10652  10658*  10660*  10662*
                          10663*  10664*  10665  10683*  10742  10753*
R6      =X000006            78*    80    1135*  1136*  1137
R7      =X000007            79*    81
SAVDT   001776            1052*  1489*  2225   2227   3434   3436
SAVER   041214            2308   2323   2388   2421   3040   3058   3125   3193   3535   3554   3604   3646   3664
                          3714   3766   3796   3842   3921   3938   3986   4044   4069   4114   4176   4201   4246
                          4300   4317   4388   4548   6675   6699   7354   7379   7501   7521   7612   7631   7718
                          7737   8222*
SAVSN   002000            1055*  1488*  2237   2239   3445   3447
SC      = 100000           780*  1596   1599   2479   4537   6207   6209   7368   7513
SCOP1   = 104420          10727*
SCRC    050366            9870*  9877*
SCYL    050356            9866*  9873*  9951
SC1     = 000100           868*
SC10    = 001000           871*
SC2     = 000200           869*
SC20    = 002000           872*
SC4     = 000400           870*
SEARCH  051204            8690   9243   9744   10043*
SECGAP  046772            4609   4720   4840   4964   9700*  9879
SECOTR  045154            5080*  5081*  5207*  5208*  5333*  5334*  5450*  5451*  5546*  5547*  6061*  6062*  6171*
                          6172*  6425*  6426*  6555*  6556*  7209*  7210*  7322*  7323*  8247*  8248*  8345*  8346*
                          8506*  8507*  9240   9252*
SECTR   051202            4408*  4430   4500   10041*  10043*  10051
SEECOM  002054            1099*  3524   3643   3901   8717
SEGPER  050306            9748*  9764*  9914*  9925*
SELECT  001770            1047*  1128*  1130*  1207   1442   7793
SELTST  007256            1359   1442*
SERCH   002036            1092*  4385
SETDSK  042204            6770   6850   6930   7005   7065   7133   8486*
SILOTB  047070            2579   2599   2604   9706*
SKEY1   050362            9868*  9875*
SKEY2   050364            9869*  9876*
SKI     = 040000           945*
SN      001730            1033*
SND1    005240            1162   1206*
SP      =%000006            80*  1139*  1215*  1216   1217   1227*  1239   1254*  1274*  1415*  1421*  1430*  1469*
                          1476*  1483*  1535*  1560*  1582*  1605*  1606*  1607   1636*  1660*  1684*  1714*  1715*
                          1716   1717   1766*  1792*  1817*  1842*  1863*  1898*  1927*  1928*  1929*  1934   1943*
                          1944*  1945   1977*  1990*  1998*  2112*  2113*  2114   2214*  2215   2216*  2217   2299*
                          2367*  2400*  2401*  2402   2412*  2413*  2414*  2416   2464*  2468*  2469*  2471   2498*
                          2499*  2500   2521*  2524*  2525*  2527   2539*  2540*  2542   2591*  2592*  2594   2606*
                          2607   2610   2625*  2626*  2627   2629*  2632   2679*  2680*  2682   2699*  2700*  2702
                          2734*  2735*  2736   2754*  2757*  2758*  2759*  2791*  2792*  2793*  2794*  2795*  2797
                          2805*  2806*  2809   2810*  2811*  2812   2814   2833*  2837*  2838*  2839*  2865*  2870*
```

2871* 2872* 2925* 2968* 2969* 2970 3022* 3034* 3035* 3036 3236* 3269* 3270*
3271 3317* 3318* 3319 3424* 3425 3426* 3427 3512* 3525* 3526 3754* 3789*
3885* 4028* 4159* 4289* 4354* 4376* 4432* 4433* 4434 4502* 4503* 4504 4581*
4606* 4717* 4837* 4961* 5054* 5061* 5064 5067 5095* 5096* 5097 5140* 5141*
5142 5181* 5188* 5191 5194 5222* 5223* 5224 5267* 5268* 5269 5307* 5314*
5317 5320 5348* 5349* 5350 5393* 5394* 5395 5438* 5470* 5471* 5472 5509*
5510* 5511 5530* 5562* 5563* 5564 5623* 5624* 5625 5651* 5702* 5703* 5705
5784* 5833* 5834* 5836 5916* 5956* 5957* 5958 5976* 5977* 5978 6012* 6013*
6014 6045* 6077* 6078* 6079 6136* 6137* 6138 6159* 6188* 6189* 6190 6214*
6215* 6216 6260* 6313* 6314* 6316 6399* 6406* 6409 6412 6440* 6441* 6442
6485* 6486* 6487 6543* 6611* 6612* 6613 6650* 6737* 6763* 6817* 6843* 6897*
6923* 6972* 6998* 7058* 7126* 7175* 7197* 7229* 7230* 7231 7268* 7269* 7270
7277* 7278* 7279 7312* 7341* 7342* 7343 7427* 7428* 7429 7441* 7462* 7568*
7679* 7784* 7790* 7798* 7840* 7891* 7897* 7913* 7914 7924 7925* 7942* 7943*
7944* 7951 7952 7953 8011* 8012* 8013 8036* 8037* 8038* 8048 8049 8050
8057* 8089 8113* 8114* 8115 8134* 8141 8170* 8180* 8181 8186* 8187 8223*
8224* 8225* 8232 8233 8234 8263* 8264* 8265 8296* 8297* 8298* 8299* 8300*
8322* 8323* 8324 8329 8330 8331 8332 8333 8363* 8364* 8365 8409* 8411*
8412* 8413* 8414* 8466 8467 8468 8469 8470 8487* 8488* 8489* 8514 8515
8516 8544* 8546* 8548 8674* 8675* 8692 8693 8711* 8733 8822* 8823* 8824*
8825* 8826* 8827* 8930* 8931* 8932 8962 8963 8964 8965 8966 8967 8985*
9026 9040* 9041* 9042* 9043* 9044* 9045* 9065 9066 9067 9068 9069 9070
9087* 9098 9105* 9112 9125* 9131* 9223 9225* 9226* 9227* 9228* 9229* 9230*
9240* 9241* 9242 9306 9307 9308 9309 9310 9311 9364* 9460 9484* 9499*
9502* 9503 9516* 9519* 9520 9529 9552* 9553* 9554* 9555* 9597 9598 9599
9600 9627* 9628* 9629* 9630* 9631* 9639* 9640* 9641 9642 9671 9672 9673
9674 9675 9725 9727* 9728* 9729* 9730* 9731* 9732* 9741* 9742* 9743 9841
9842 9843 9844 9845 9846 9878* 9999 10044* 10045* 10046* 10047* 10048* 10049*
10114 10115 10116 10117 10118 10119 10134* 10154 10202* 10205 10207 10208 10236
10237 10241* 10258* 10259* 10260* 10261* 10262* 10263* 10264 10267* 10280 10282* 10284
10294 10296 10298 10299 10300 10301 10302 10304* 10305* 10338* 10339 10340* 10342
10343 10344* 10347 10349* 10351* 10357 10390* 10391* 10392 10400 10403 10416* 10417*
10418* 10423* 10436* 10441 10452 10453* 10454* 10455* 10477* 10478* 10479* 10480* 10481*
10483 10487* 10489 10491 10499* 10500 10502 10503* 10505 10506 10507 10509 10541
10553* 10556* 10569* 10574* 10593* 10599* 10603* 10610 10611 10642* 10643 10644 10645*
10650* 10651* 10652* 10658 10683 10684 10685 10686* 10687* 10703* 10704 10737* 10738*
10739* 10740* 10741* 10742* 10743 10749* 10753 10754 10755 10756 10757 10758 10763*

SPACE? 057456 1981 11088*
SPACE8 057450 1975 1976 1984 1985 1986 11087*
SR0 = 177572 192*
SR1 = 177574 193*
SR2 = 177576 194*
SR3 = 172516 195*
SSECTR 050360 9867* 9874*
SSYN 045164 9246* 9258* 9392*
STACK = 001000 61* 1139 1227 1274 1535 1560 1582 1636 1660 1684 1766 1792 1817
1842 1863 1898 2299 2367 2464 2521 2754 2833 2865 2925 3022 3236
3512 3754 3885 4028 4159 4289 4354 4376 4581 4606 4717 4837 4961
5054 5181 5307 5438 5530 5651 5784 5916 6045 6159 6260 6399 6543
6650 6737 6763 6817 6843 6897 6923 6972 6998 7058 7126 7175 7197
7312 7441 7462 7568 7679
START 004224 1129 1133*
STKLMT= 177774 66*
SWR = 177570 68* 69 1242 1316 1357 1971 2625 3781 5509 5623 6136 6611 7268

```
                        7427   7985   8055   8322   8923   8925  10197*  10211  10213  10219  10226  10537  10544
                       10548  10551
SW0    = 000001          119*
SW00   = 000001          109*   119
SW01   = 000002          108*   118
SW02   = 000004          107*   117
SW03   = 000010          106*   116
SW04   = 000020          105*   115
SW05   = 000040          104*   114
SW06   = 000100          103*   113
SW07   = 000200          102*   112   2626   2627   5511   5625   6138   6613   7270   7429   8324
SW08   = 000400          101*   111   2626
SW09   = 001000          100*   110   7985   8055
SW1    = 000002          118*
SW10   = 002000           99*
SW11   = 004000           98*
SW12   = 010000           97*
SW13   = 020000           96*  1242   1316   1357   1971   3781
SW14   = 040000           95*
SW15   = 100000           94*
SW2    = 000004          117*
SW3    = 000010          116*
SW4    = 000020          115*
SW5    = 000040          114*
SW6    = 000100          113*  8925
SW7    = 000200          112*
SW8    = 000400          111*  8923
SW9    = 001000          110*
TAGDTE  002016          1074*
TBITVE= 000014           152*
TDF    = 000040           88*
TESDTE  002014          1070*  9404   9937  10195*
TESTAD  037344          7883*
TIMCNT  041072          8179*  8188   8193
TKVEC  = 000060           159* 10381* 10382*
TMPILL  002010          1064*  2929*  2932   2934*  2939   2941   2942*
TOLGAP  050076          5443   6551   7202   7317   9634   9711*
TOTALA  002006          1062*  1350*  1352*  3259   3404
TPVEC  = 000064           160*
TRAPVE= 000034           158*  1144*  1145*
TRE    = 040000           77*  2479   2488   2998   6207   6209   7368   7513
TRK     045124          9242*  9244*
TRK1   = 004000           87*
TRK10  = 040000           87*
TRK2   = 010000           87*
TRK20  = 100000           87*
TRK4   = 020000           87*
TRTVEC= 000014           153*
TSECC   002012          1066*  9643  10135  10145  10187*
TSECCG  043222          8792*  8916   9492   9504   9509   9521   9571   9586   9645*  9657*  10137*  10147*  10191*
TSTNM   002020          1077*  1228*  1275*  1296*  1502*  1536*  1561*  1583*  1637*  1661*  1685*  1707*  1767*
                        1793*  1818*  1843*  1864*  1899*  2300*  2368*  2465*  2522*  2578*  2643*  2677*  2753*
                        2788*  2832*  2866*  2926*  3023*  3099*  3237*  3513*  3755*  3886*  4029*  4160*  4290*
                        4377*  4607*  4718*  4838*  4962*  5055*  5182*  5308*  5435*  5439*  5532*  5652*  5785*
```

DZRPSD.P11 CROSS REFERENCE TABLE

		5917#	6047#	6161#	6262#	6401#	6545#	6651#	6764#	6844#	6924#	6999#	7059#	7127#
		7198#	7313#	7463#	7569#	7680#	7891	11500	11502	11504	11507	11510	11513	11515
		11518	11522	11525	11527	11528	11531	11534	11538	11541	11544	11548	11551	11554
		11557	11560	11565										
TST1	005346	1208	1225#	7854										
TST10	010136	1558#												
TST100	033354	6760#												
TST101	033476	6816#												
TST102	033524	6840#												
TST103	033646	6896#												
TST104	033674	6920#												
TST105	033774	6971#												
TST106	034022	6995#												
TST107	034122	7055#												
TST11	010202	1580#												
TST110	034250	7123#												
TST111	034354	7174#												
TST112	034402	7195#												
TST113	035012	7311#												
TST114	035542	7440#												
TST115	035570	7461#												
TST116	036122	7567#												
TST117	036446	7678#												
TST12	010352	1634#												
TST120	036760	7777#												
TST13	010416	1658#												
TST14	010462	1682#												
TST15	010526	1705#												
TST16	010714	1763#												
TST17	010760	1789#												
TST2	005622	1272#												
TST20	011024	1815#												
TST21	011070	1840#												
TST22	011134	1861#												
TST23	011200	1897#												
TST24	011546	2049#												
TST25	012572	2278	2298#											
TST26	012734	2366#												
TST27	013304	2463#												
TST3	005674	1293#												
TST30	013450	2501	2520#											
TST31	013664	2559	2577#											
TST32	014112	2642#												
TST33	014224	2660	2675#											
TST34	014512	2733	2751#											
TST35	014632	2771	2786#											
TST36	014776	2816	2830#											
TST37	015122	2850	2863#											
TST4	005746	1301	1311#											
TST40	015246	2883	2923#											
TST41	015542	3021#												
TST42	015766	3098#												
TST43	016356	3235#												
TST44	017510	3484	3511#											

```
TST45   020364          3753#
TST46   021024          3884#
TST47   021432          4027#
TST5    007272          1443    1455#   7801    7810
TST50   021776          4158#
TST51   022342          4288#
TST52   022520          4353#
TST53   022546          4375#
TST54   023434          4580#
TST55   023462          4605#
TST56   023772          4672    4695    4716#
TST57   024330          4782    4810    4836#
TST6    007530          1501#
TST60   024670          4906    4933    4960#
TST61   025134          5053#
TST62   025520          5134    5159    5180#
TST63   026104          5261    5286    5306#
TST64   026470          5387    5412    5433#
TST65   027030          5491    5512    5529#
TST66   027374          5585    5626    5644#
TST67   027724          5754    5776#
TST7    010072          1505    1533#
TST70   030232          5880    5914#
TST71   030622          6024    6044#
TST72   031166          6139    6158#
TST73   031520          6252#
TST74   032070          6371    6396#
TST75   032472          6479    6518    6539#
TST76   033022          6621    6648#
TST77   033326          6736#
TTNO  = 000117          1228#   1275#   1296#   1502#   1536#   1561#   1583#   1637#   1661#   1685#   1707#   1767#   1793#
                        1818#   1843#   1864#   1899#   2300#   2368#   2465#   2522#   2578#   2643#   2677#   2753#   2788#
                        2832#   2866#   2926#   3023#   3099#   3237#   3513#   3755#   3886#   4029#   4160#   4290#   4377#
                        4607#   4718#   4838#   4962#   5055#   5182#   5308#   5435#   5439#   5532#   5652#   5785#   5917#
                        6047#   6161#   6262#   6401#   6545#   6651#   6764#   6844#   6924#   6999#   7059#   7127#   7198#
                        7313#   7463#   7569#   7680#
TUF   = 000100           887#
TY      045270          9353#
TYPDS = 104410          1416    1431    3790    7785    7791    7799    7841    10604   10723#
TYPE  = 104400          1165    1169    1173    1177    1181    1185    1189    1193    1197    1201    1210    1245    1249
                        1319    1323    1327    1331    1335    1363    1367    1371    1375    1379    1410    1417    1423
                        1429    1464    1471    1472    1478    1479    1485    1506    1510    1514    1520    1975    1976
                        1981    1984    1985    1986    1995    3495    3784    4395    4644    4674    4757    4784    4879
                        4908    4998    5105    5136    5232    5263    5358    5389    5476    5570    5920    6085    6196
                        6450    6481    6578    6654    7235    7347    7490    7596    7702    7780    7786    7797    7800
                        7839    7842    7887    7893    7899    7900    7904    7908    7915    7919    8271    8371    8538
                        9083    9089    9101    9107    9113    9117    9121    9127    9133    10161   10303   10394   10399
                        10444   10447   10451   10511   10513   10539   10547   10568   10585   10587   10590   10592   10607
                        10612   10677   10719#  10761
TYPOC = 104402          1256    1422    1470    1477    1484    7892    7898    9088    9106    9126    9132    10165   10576
                        10600   10720#
TYPON = 104406          10722#
TYPOS = 104404          1978    1991    10721#
T.SCOP  040504          8055#   10727
```

```
UN      005664          1277*  1278*  1280#
UNIT    001762          1042#  1216*  1438#  1444#  1461  1469  1539  1564  1605  1640  1664  1688  1770
                        1796   1821   1846   1867   2071  2087  2113  2396  2400  2468  2498  2524  2539
                        2591   2679   2699   2734   2794  2800  2805  2810  3168  3269  3318  3588  3697
                        3789   3828   3971   4101   4233  5702  5833  5945  6214  6313  6660  7784  7804
                        7809*  7990   8077
UNITS   001742          1041#  1388   1393   1438   1953  1964  7805
UNITSL  001772          1048#  1217*  1444
UNLOAD  002026          1088#  3762
UNS     = 040000         816#
UPE     = 020000         754#  2807   2811
US1     = 000001         741#
US2     = 000002         742#
US4     = 000004         743#
UWR     = 000010         941#
VUF     = 000002         940#
VU30    = 010000         893#
VV      = 000100         790#  2216   2317   2970   3052  5957  5977  6014  7279  7282  8114
WAIT.T  041074          8180#  10729
WAT     = 104424        2536   2695   2993   3114   3117  3182  3185  7290  10729#
WC      001674          1017#  3106   3140   3174   3208  6329  7946  11518
WCE     = 040000         755#  5707   5838   6314   6318
WCF     = 000040         807#
WCRC    047052          5088   5215   5341   5458   5554  6069  6179  6433  7066*  7134*  7217  7330  8255
                        8354   8513   9402   9703#
WCU     = 000001         881#
WCYL    052274          4619#  4626   4729*  4736   4850*  4857  4974*  4981  6562  9756#
WECC1   050070          4676#  4786*  4910*  9708#  11551
WECC2   050072          4677#  4787*  4911*  9709#  11551
WKEY1   050300          4622#  4732*  4853*  4977*  9758#
WKEY2   050302          4623#  4733*  4854*  4978*  9759#
WLE     = 004000         813#
WORD    046062          9382#  9386*  9394*  9396*  9398*  9400*  9402*  9443*  9447*  9477*  9487*  9500*  9517*
                        10141* 10149*
WRCHDA  041560          5826   6306   8344#
WRCHDT  002042          1094#  8274
WRCHEK  002040          1093#  8374
WRCHMD  041246          5696   8246#
WRDATA  046540          9297   9622#  9836
WRFROM  002070          1109#  2325   2331*  2339   2423  2429*  2434  3060  3066*  3073  3127  3133*  3141
                        3195   3201*  3209   3556   3562* 3569   3606  3613*  3619  3666  3673*  3681  3716
                        3722*  3729   3798   3804*  3812  3844   3851* 3856  3940  3946*  3954  3988  3994*
                        3999   4071   4077*  4084   4116  4122*  4128  4203  4209*  4216  4248  4254*  4260
                        4319   4325*  4332   4550   4556* 4564   4632  4742  4863  4987  5138  5154  5265
                        5281   5391   5407   5464   5465  5469   5587  5588  5592  5593  5598  5722  5729
                        5730   5734   5741   5742   5749  5854   5860  5867  5868  5874  6102  6103  6107*
                        6111   6345   6351   6358   6359  6365   6483  6499  6568  6569  6573  6667  6701
                        6707*  6714   6781   6790   6791  6861   6870  6871  6935  6936  6959  7010  7011
                        7034   7077   7086   7087   7139  7140   7163  7223  7224  7228  7335  7336  7340
                        7364   7381   7387*  7394   7523  7529*  7536  7581  7633  7639*  7646  7692  7739
                        7745*  7752   8579   8679
WRHEAD  050370          9755   9873#
WRIDAT  002044          1095#  5479   6581   7238   7350  7706
WRIFOR  002046          1096#  4648   4761   4883   5002  6664  7600  8676
```

```
WRITE   046320          9411    9422    9551#   9647    9653    9659    9663    9666    9919    9939    9953    9966    9978
                        9989
WRL   = 004000           795#
WRU   = 000400           889#
WSECTR  050276          4621#   4731#   4852#   4976#   9741    9757#
WSSYNC  047040          9701#
WSU   = 000004           883#
WTRK    050232          9743#   9745#
WWORD   046316          9412    9416    9419    9423    9430    9435    9439    9546#   9595#   9648    9654    9660    9664
                        9667    9920    9924    9941    9954    9967    9979    9990
X       045162          5085#   5212#   5338#   5455#   5551#   6066#   6176#   6430#   6560#   7214#   7327#   8252#   8350#
                        8552#   8569    9255#   9290
XE2     007002          1355    1386#
Y       045222          9300#
ZCODE   043236          8804#   8997#   11560
ZER   = 000400           828#   9006
ZWORDS  046536          9621#   9641#
$BDADR  001122           246#   2473#   2482#   2494#   2502#   2529#   2544#   2596#   2655#   2684#   2704#   2951#   2962#
                        2977#   3001#   5015#   8098#   8104#   8108#   8120#   8125#   8128#   8147#   8154#   8160#   8165#
                       11507   11510   11531
$BDDAT  001126           248#   1299#   1300    1595#   1596    1608#   1609    1615#   1616    1719#   1741#   1968#   1988
                        1990   2078#   2079    2093#   2094    2103#   2104    2115#   2116    2126#   2127    2136#   2137
                        2146#   2147   2156#   2157    2166#   2167    2176#   2177    2186#   2187    2194#   2196    2205#
                        2206   2215#   2226#   2227    2238#   2239    2249#   2250    2260#   2261    2554#   2561#   2609#
                        2657#   2659    2719#   2727#   2737#   2760#   2761    2769#   2770    2796#   2797    2813#   2814
                        2840#   2841    2848#   2849    2873#   2874    2881#   2882    3284#   3285    3295#   3296    3306#
                        3307   3320#   3321    3335#   3337    3346#   3347    3356#   3357    3366#   3367    3376#   3377
                        3386#   3387    3396#   3397    3406#   3407    3415#   3416    3425#   3435#   3436    3446#   3447
                        3456#   3457    3466#   3467    4425#   4427    4495#   4498    4666#   4776#   4898#   5123#   5250#
                        5376#   5500#   5605#   5972#   6118#   6205#   6207    6217#   6218    6223#   6225    6468#   6599#
                        7259#   7283#   7419#   7998#   8000#   8002    8004#   8198#   8314#   8622#   8623    8632#   8633
                        8644#   8645    8652#   8654    8725#   8726    9416#   9435#   9913#   9924#   9946#   9960#   9970#
                        9983#   9994#  11500   11504   11510   11515   11528   11534   11538   11541   11544   11565
$BELL   001210           274#  10399   10460   10539   10559
$CMTAG  001100           234#   1134    1135    1142    1148    1149
$CM1  = 000006           260#    261#    262#    263#    264#    265#    266#
$CM2  = 000014           260#    261#    262#    263#    264#    265#    266#
$CM3  = 000006           258#    260
$CM4  = 000006           266#    267#    268#    269#    270#    271#    272#
$CNTLC  053073         10394   10460#
$CRLF   001215           276#   1429    1471    1478    1485    1995    7899   10460   10516   10547   10559   10568   10587
                       10592   10612
$DBLK   052354         10269   10303   10311#
$DOAGN  037320          7835    7844    7854#
$DTBL   052344         10272   10307#
$ENDAD  037310           228     230    7845    7848    7850#
$ENDCT  037240          7837#
$ENDMG  037324          7797    7839    7855#
$ENULL  037341          7800    7842    7858#
$EOP    037204          1257    1384    7803    7827#
$EOPCT  037232          7834#   7838
$ERFLG  001103           237#   7987   10176   10215   10217   10223#  10244   10534#  10559
$ERMAX  001115           243#   1150#  10217   10239#  10244
$ERROR  053240          1142    1258   10531#
```

```
$ERRPC  001116          244#  10541#  10542#  10543   10559   10574   11500   11502   11504   11507   11510   11513   11515
                      11518   11522   11525   11527   11528   11531   11534   11538   11541   11544   11548   11551   11554
                      11557   11560   11565
$ERRTB  001220          294#  10582
$ERRTY  053402        10546   10567#
$ERTTL  001112          241#   7790    7792#  10548#  10559
$ESCAP  001206          273#   1149#  10238#  10555.  10556   10559
$FILLC  001150          256#  10347   10359
$FILLS  001147          255#  10359
$GDADR  001120          245#
$GDDAT  001124          247#   1302#   1599#   1607#   1609   1619#   1716#   1740#   1966#   1973   1977   2076#   2079
                       2092#   2102#   2114#   2116   2125#   2135#   2145#   2155#   2165#   2175#   2185#   2195#   2196
                       2204#   2213#   2225#   2237#   2248#   2259#   2552#   2560#   2610#   2658#   2659   2718#   2726#
                       2736#   2763#   2772#   2800#   2801#   2812#   2843#   2851#   2876#   2884#   3283#   3294#   3305#
                       3319#   3321   3332#   3334#   3337   3345#   3355#   3365#   3375#   3385#   3395#   3404#   3405#
                       3407   3414#   3423#   3434#   3445#   3455#   3465#   5499#   5604#   5971#   6117#   6209#   6216#
                       6218   6227#   6598#   7258#   7282#   7418#   7993#   7995#   7997   8002   8007#   8013#   8313#
                       8627#   8638#   8648#   8657#   8728#   9391#   9415#   9432#   9434#   9457#   9912#   9923#   9945#
                       9959#   9971#   9982#   9995#  11500   11502   11504   11534   11538   11541   11544   11548
$GET42  037262          7843#
$HD   = 000000            11
$HIOCT  053236        10504#  10515#
$ICNT   001104          238#  10230#  10231   10233#  10243
$ILLUP  054200        10735   10766#
$ITEMB  001114          242#  10543#  10559   10571
$LF     001216          277#  10451   10460   10516   10559
$LPADR  001106          239#   1151#   1457#   7914#   7925   10221#  10236#  10241   10243
$LPERR  001110          240#   1152#   2947#   3158#   5937#   6002#   7897    7924#  10221   10237#  10243   10553
$MXCNT  052136        10234   10243#
$NULL   001146          254#  10349   10359
$NWTST= 000001          1221#   1223    1265#   1267   1290#   1308#   1448#   1450   1493#   1495   1527#   1529   1552#
                        1554   1574#   1576   1628#   1630   1652#   1654   1676#   1678   1699#   1701   1757#   1759
                        1783#   1785   1809#   1811   1834#   1836   1855#   1857   1889#   1891   2038#   2040   2289#
                        2291   2358#   2360   2453#   2455   2506#   2508   2569#   2571   2636#   2638   2667#   2669
                        2745#   2747   2780#   2782   2824#   2826   2857#   2859   2916#   2918   3012#   3014   3090#
                        3092   3225#   3227   3502#   3504   3745#   3747   3875#   3877   4018#   4020   4149#   4151
                        4279#   4281   4350#   4364#   4366   4577#   4595#   4597   4706#   4708   4827#   4829   4951#
                        4953   5045#   5047   5172#   5174   5298#   5300   5424#   5426   5520#   5522   5635#   5637
                        5766#   5768   5900#   5902   6034#   6036   6148#   6150   6240#   6242   6387#   6389   6530#
                        6532   6630#   6632   6733#   6747#   6749   6813#   6827#   6829   6893#   6907#   6909   6968#
                        6982#   6984   7042#   7044   7110#   7112   7171#   7186#   7188   7297#   7299   7437#   7451#
                        7453   7554#   7556   7665#   7667   7768#   7770
$OCNT   054000        10649#  10678#  10691#
$OMODE  054002        10644#  10648#  10653   10656#  10667#  10693#
$OVER   052122        10198   10214   10222   10232   10240#
$PASS   001100          235#   3779    7796#   7798   7831#   7832#   7840    7855   10228   10244
$POWER  054206        10762   10769#
$PWRAD  054174        10764#
$PWRON  054052          1146  10735#  10759
$PWRMG  054170        10762#
$PWRUP  054120        10744   10749#
$QUES   001214          275#  10444   10460   10513   10516   10559
$RDCHR  052670        10416#  10724
$RDDEC= ****** U      10727
```

```
$RDLIN  052754          10436#  10725
$RDOCT  053100          10477#  10726
$RDSZ = 000011          10429#
$REGAD  001152            258#
$REGO   001154            260#
$REG1   001156            261#
$REG2   001160            262#
$REG3   001162            263#
$REG4   001164            264#
$REG5   001166            265#
$RESET  037306           7846    7849#
$SAVRE= ****** U        10727
$SAVR6  054204          10743#  10749   10750#  10751#  10768#
$SCOPE  051652           1140   10185#
$SETUP= 000017           1134#   1140    1142    1144    1146    1148    1149    1151    7829
$SS1  = 000000           1154#
$STUP = 177777           1134#
$SVLAD  052074          10206   10235#
$SWR  = 167700              1#     11      47      48      49      50      51      52      53     272     273     274    1148
                         1149    1151    1152    1226    1273    1294    1312    1456    1502    1534    1559    1581    1635
                         1659    1683    1706    1764    1790    1816    1841    1862    1898    2050    2299    2367    2464
                         2521    2578    2643    2676    2752    2787    2831    2864    2924    3022    3099    3236    3512
                         3754    3885    4028    4159    4289    4354    4376    4581    4606    4717    4837    4961    5054
                         5181    5307    5434    5530    5645    5777    5915    6045    6159    6253    6397    6540    6649
                         6737    6761    6817    6841    6897    6921    6972    6996    7056    7124    7175    7196    7312
                         7441    7462    7568    7679    7778    7824    7830    7843    7855   10177   10178   10179   10180
                        10181   10197   10209   10211   10212   10215   10216   10217   10224   10225   10226   10237   10240
                        10243   10523   10524   10525   10526   10527   10537   10544   10548   10551   10559
$SWRMK= 000000             53      54   10181   10182   10213
$TIMES  001204            272#   1148#   1226#   1273#   1294#   1312#   1456#   7778#   7830#  10224#  10231   10234#  10243
$TKB    001140            251#  10363   10383   10390
$TKCNT  052472          10363#  10378#  10397   10402#  10420   10422#
$TKINT  052512           1158    1315    2395    7991    9093   10378#  10395
$TKQEN= 052511          10367#  10405   10425
$TKQIN  052474          10364#  10379#  10380   10403#  10404#  10405   10407#
$TKQOU  052476          10365#  10380#  10423   10424#  10425   10427#
$TKQSR  052500          10366#  10379   10407   10427
$TKS    001136            250#   1137   10363   10384#
$TKSRV  052562          10381   10390#
$TMP0   001170            266#   1397#   1398#   8184#   8189    8194    8198   8304#   8318   8417#   8463#  11515
$TMP1   001172            267#   1240#   4402#   4427    4494#   4498    4512#   4542   5939#   5994#   8185#   8189    8194
                         8305#   8320   11513   11515   11528
$TMP2   001174            268#   4430#   4431#   4500#   4501#   8188#   8191#  8193#   8196#   8419#   8461#  11528
$TMP3   001176            269#   4434#   4504#   8182#   8183#   8418#   8420   8421#  11515   11528
$TMP4   001200            270#   4424#   4433    4455#   4484#   4503    8420#  8444#
$TMP5   001202            271#   3528#   3530#   3531    3598    8416#   8422#  8428    8541#   8620    8642
$TN   = 000121              1#     11    1208    1221    1226#   1228    1265   1273#   1275    1290   1294#   1296    1301
                         1308   1312#   1443    1448    1456#   1493    1502#  1505    1527   1534#   1536    1552   1559#
                         1561    1574   1581#   1583    1628    1635#   1637   1652    1659#   1661    1676   1683#   1685
                         1699   1706#   1707    1757    1764#   1767    1783   1790#   1793    1809   1816#   1818    1834
                         1841#   1843    1855   1862#   1864    1889    1898#  1899    2038   2050#   2277    2289   2299#
                         2300    2358   2367#   2368    2453    2464#   2465   2501    2506   2521#   2522    2559    2569
                         2578#   2636   2643#   2660    2667    2676#   2677   2733    2745   2752#   2753    2771    2780
                         2787#   2788    2816    2824   2831#   2832    2850   2857   2864#   2866    2883    2916   2924#
```

```
                    2926    3012    3022#   3023    3090    3099#   3225    3236#   3237    3483    3502    3512#   3513
                    3745    3754#   3755    3875    3885#   3886    4018    4028#   4029    4149    4159#   4160    4279
                    4289#   4290    4350    4354#   4364    4376#   4377    4577    4581#   4595    4606#   4607    4672
                    4695    4706    4717#   4718    4782    4810    4827    4837#   4838    4906    4933    4951    4961#
                    4962    5045    5054#   5055    5134    5159    5172    5181#   5182    5261    5286    5298    5307#
                    5308    5387    5412    5424    5434#   5435    5439    5491    5512    5520    5530#   5532    5585
                    5626    5635    5645#   5652    5754    5766    5777#   5785    5880    5900    5915#   5917    6024
                    6034    6045#   6047    6139    6148    6159#   6161    6240    6253#   6262    6371    6387    6397#
                    6401    6479    6518    6530    6540#   6545    6621    6630    6649#   6651    6733    6737#   6747
                    6761#   6764    6813    6817#   6827    6841#   6844    6893    6897#   6907    6921#   6924    6968
                    6972#   6982    6996#   6999    7042    7056#   7059    7110    7124#   7127    7171    7175#   7186
                    7196#   7198    7297    7312#   7313    7437    7441#   7451    7462#   7463    7554    7568#   7569
                    7665    7679#   7680    7768    7778#
$TPB    001144          253#  10357#  10359
$TPFLG  001151          257#  10334   10359
$TPS    001142          252#  10355   10359
$TRAP   054004         1144   10703#
$TRP  = 000026        10710#  10720#  10721#  10722#  10723#  10724#  10725#  10726#  10727#  10728#  10729#  10730#
$TRPAD  054024        10707   10718#
$TSTNM  001102          236#   7795#   7829#  10176   10213   10235#  10240   10244   10536   10559
$TTYIN  053062        10437   10438   10455   10459#
$TYPBN= ****** U      10724
$TYPDS  052140        10257#  10723
$TYPE   052364        10334#  10710   10719
$TYPEC  052454        10346   10353   10355#  10356
$TYPOC  053602        10647#  10720
$TYPON  053616        10646   10649#  10722
$TYPOS  053556        10642#  10721
$XTSTR  051700        10200#
$STRP = 000002        10709#  10720   10721   10722   10723   10724   10725   10726   10727   10728   10729   10730
$OFILL  054001        10643#  10647#  10657   10692#
.     = 065000          166#    170     172#    175#    220#    229#    232#    278    1041#   1109#   1110#   1138    1151
                       1152    1165    1168#   1169    1173    1177    1181    1185   1188#   1189    1192#   1193    1197
                       1201    1210    1245    1248#   1249    1319    1323    1327    1331   1334#   1335    1363    1366#
                       1367    1371    1374#   1375    1379    1382#   1410    1413#   1417   1420#   1423    1426#   1464
                       1467#   1472    1479    1506    1509#   1510    1513#   1514    1520   1523#   3784    3787#   7780
                       7786    7789#   7855    7859    7887    7893    7900    7903#   7904   7908    7915    7918#   7919
                       9083    9089    9101    9104#   9107    9113    9117    9121    9127   9133    9700#   9701#   9702#
                       9703#   9704#   9705#   9707#   9708#   9709#   9710#   9711#  10161  10243   10244   10311#  10359
                      10363   10366#  10367   10368#  10459#  10460   10461   10516  10559  10615#  10746   10767   11498#
                      11628#
```

DZRPSD.P11 CROSS REFERENCE TABLE

```
CHECKA      41#  4673  4783  4907  5135  5262  5388  6480
CHECKB      41#  3494  4394  4643  4756  4878  4997  5104  5231  5357  5475  5569  5919  6084  6195
            6449  6577  6653  7234  7346  7489  7595  7701  8270  8370  8537
COMMEN      13   162#  2005
ENDCOM      20   162#  2030
ERROR       62#  1241  1282  1304  1543  1568  1601  1612  1621  1644  1668  1692  1721  1743  1774
            1800  1825  1850  1871  1962  2081  2282  2349  2443  2474  2483  2495  2503  2530  2545
            2556  2563  2597  2614  2616  2656  2662  2685  2705  2720  2728  2739  2765  2774  2803
            2818  2845  2853  2878  2886  2952  2963  2978  3002  3082  3150  3218  3487  3579  3629
            3689  3739  3821  3866  3963  4009  4093  4139  4225  4270  4341  4435  4505  4572  4668
            4696  4698  4778  4811  4813  4900  4934  4936  5016  5125  5160  5162  5252  5287  5289
            5378  5413  5415  5504  5507  5609  5611  5709  5715  5755  5757  5840  5845  5882  5884
            5951  5960  5973  5980  6008  6018  6025  6122  6124  6211  6221  6229  6320  6325  6332
            6338  6373  6375  6470  6505  6507  6519  6603  6609  6622  6722  7263  7266  7284  7403
            7423  7425  7545  7655  7761  8099  8105  8109  8121  8126  8129  8148  8155  8161  8166
            8199  8585  8587  8611  8613  8628  8639  8649  8659  8730  8936  8942  9010
ESCAPE      162#
HCOMPR      41#  6762  6842  7057
HCOMPW      41#  6922  6997  7125
MAKECL      41#  4350  4577  6733  6813  6893  6968  7171  7437
MSG         1221#  1223  1264#  1267  1447#  1450  1493#  1495  1527#  1529  1551#  1554  1573#  1576  1627#
            1630  1651#  1654  1675#  1678  1698#  1701  1756#  1759  1782#  1785  1808#  1811  1833#  1836
            1854#  1857  1888#  1891  2036#  2040  2288#  2291  2357#  2360  2452#  2455  2505#  2508  2568#
            2571  2635#  2638  2666#  2669  2744#  2747  2779#  2782  2823#  2826  2856#  2859  2915#  2918
            3011#  3014  3089#  3092  3224#  3227  3501#  3504  3744#  3747  3874#  3877  4017#  4020  4148#
            4151  4278#  4281  4363#  4366  4594#  4597  4705#  4708  4826#  4829  4950#  4953  5044#  5047
            5171#  5174  5297#  5300  5423#  5426  5519#  5522  5632#  5637  5765#  5768  5898#  5902  6033#
            6036  6147#  6150  6239#  6242  6386#  6389  6529#  6532  6629#  6632  7185#  7188  7296#  7299
            7450#  7453  7553#  7556  7664#  7667  7768#  7770
MSGA        6747#  6749  6829  7044
MSGB        6906#  6909  6984  7112
MULT        162#
NEWTST      162#  1221  1265  1290  1308  1448  1493  1527  1552  1574  1628  1652  1676  1699  1757
            1783  1809  1834  1855  1889  2038  2289  2358  2453  2506  2569  2636  2667  2745  2780
            2824  2857  2916  3012  3090  3225  3502  3745  3875  4018  4149  4279  4350  4364  4577
            4595  4706  4827  4951  5045  5172  5298  5424  5520  5635  5766  5900  6034  6148  6240
            6387  6530  6630  6733  6747  6813  6827  6893  6907  6968  6982  7042  7110  7171  7186
            7297  7437  7451  7554  7665  7768
POP         162#  7951  8048  8232  8328  8466  8514  8692  8732  8961  9025  9065  9305  9459  9529
            9597  9671  9840  9998  10113  10153  10298  10505  10753
PUSH        162#  7941  8035  8222  8295  8408  8411  8486  8673  8710  8821  8985  9039  9225  9364
            9483  9551  9627  9727  9878  10044  10134  10257  10479  10737
RFORGC      41#  4695  4810  4933  5159  5286  5412  5754  5080  6371
SAVE        41#  10532
SAVTST      41#  1228  1275  1296  1502  1536  1561  1583  1637  1661  1685  1707  1767  1793  1818
            1843  1864  1899  2300  2368  2465  2522  2578  2643  2677  2753  2788  2832  2866  2926
            3023  3099  3237  3513  3755  3886  4029  4160  4290  4377  4607  4718  4838  4962  5055
            5182  5308  5435  5439  5532  5652  5785  5917  6047  6161  6262  6401  6545  6651  6764
            6844  6924  6999  7059  7127  7198  7313  7463  7569  7680
SCOPE       63#  1225  1272  1293  1311  1455  1501  1533  1580  1634  1658  1682  1705  1763
            1789  1815  1840  1861  1897  2049  2298  2366  2463  2520  2577  2642  2675  2751  2786
            2830  2863  2923  3021  3098  3235  3511  3753  3884  4027  4158  4288  4353  4375  4580
            4605  4716  4836  4960  5053  5180  5306  5433  5529  5644  5776  5914  6044  6158  6252
            6396  6539  6648  6736  6760  6816  6840  6896  6920  6971  6995  7055  7123  7174  7195
```

CROSS REFERENCE TABLE

```
          7311    7440    7461    7567    7678    7777    7828
SETTRA   10710#  10720   10721   10722   10723   10724   10725   10726   10727   10728   10729
SETUP      162#  1134
SKIP        41#   162#   1208    1301    1443    1505    2277    2501    2559    2660    2733    2771    2816    2850    2883
          3483    4672    4782    4906    5134    5261    5387    5491    5512    5585    5626    6024    6139    6479    6518
          6621
SLASH      162#
SMORE       41#  10186
SPACE      162#
STARS       32      39    162#    219     222     278     723     725     763     765    1221    1224    1265    1271    1290
          1292    1308    1310    1448    1454    1493    1500    1527    1532    1552    1557    1574    1579    1628    1633
          1652    1657    1676    1681    1699    1704    1757    1762    1783    1788    1809    1814    1834    1839    1855
          1860    1879    1885    1889    1896    2017    2020    2038    2042    2289    2297    2358    2365    2453    2462
          2506    2519    2569    2576    2636    2641    2667    2674    2745    2750    2780    2785    2824    2829    2857
          2862    2891    2901    2908    2912    2916    2922    3012    3020    3090    3097    3225    3234    3502    3510
          3745    3752    3875    3883    4018    4026    4149    4157    4279    4287    4350    4352    4364    4374    4577
          4579    4595    4604    4706    4715    4827    4835    4951    4959    5020    5040    5045    5052    5172    5179
          5298    5305    5424    5432    5520    5528    5635    5643    5766    5775    5900    5913    6034    6043    6148
          6157    6240    6251    6387    6395    6530    6538    6630    6647    6733    6735    6747    6759    6813    6815
          6827    6839    6893    6895    6907    6919    6968    6970    6982    6994    7042    7054    7110    7122    7171
          7173    7186    7194    7297    7310    7437    7439    7451    7460    7554    7566    7665    7677    7768    7776
          7818    7861    7880    9157    9158    9217    9220    9314    9336    9343    9345    9470    9472    9537    9539
          9608    9611    9686    9692    9718    9720    9850    9852   10010   10012   10123   10125   10170   10244   10312
         10359   10409   10429   10462   10516   10559   10616   10694   10730
TJUMP       41#
TRMTRP   10710#
TSCLR       41#  2091    2101    2124    2134    2144    2154    2164    2174    2184    2203    2247    2258
TSCLR1      41#  2224    2236
TSCLR2      41#  3282    3293    3304    3344    3354    3364    3374    3384    3394    3413    3454    3464
TSCLR3      41#  3433    3444
TSCLR4      41#  2111
TSCLR5      41#  3316
TYPBIN     162#
TYPDEC     162#  7840    10602
TYPNUM     162#
TYPOCS     162#  1977    1990
TYPOCT     162#  1476    1483    10574   10599
TYPTXT     162#  1164    1169    1173    1177    1181    1185    1189    1193    1197    1201    1210    1245    1249    1319
          1323    1327    1331    1335    1362    1367    1371    1375    1379    1410    1417    1423    1464    1472    1479
          1506    1510    1514    1520    3784    7780    7786    7887    7893    7900    7904    7908    7915    7919    9082
          9089    9101    9107    9113    9117    9121    9127    9133   10160
WDATAR      41#  5437    7196
XREAD       41#  5057    5184    5310    5402
SSCMRE     222#   260     261     262     263     264     265
SSCMTM     222#   266     267     268     269     270     271
SSESCA     162#
SSNEWT     162#  1221    1265    1290    1308    1448    1493    1527    1552    1574    1628    1652    1676    1699    1757
          1783    1809    1834    1855    1889    2038    2289    2358    2453    2506    2569    2636    2667    2745    2780
          2824    2857    2916    3012    3090    3225    3502    3745    3875    4018    4149    4279    4350    4364    4577
          4595    4706    4827    4951    5045    5172    5298    5424    5520    5635    5766    5900    6034    6148    6240
          6387    6530    6630    6733    6747    6813    6827    6893    6907    6968    6982    7042    7110    7171    7186
          7297    7437    7451    7554    7665    7768
SSSET    10710#  10720   10721   10722   10723   10724   10725   10726   10727   10728   10729
SSSKIP     162#
```

DZRPSD.P11 CROSS REFERENCE TABLE

```
.EQUAT      1#     57
.FLOAT     41#   1534    1559    1635    1659    1683    1765    1791    1816    1841    1862
.HEADE      1#
.KT11       1#    183
.SETUP      1#   1133
.SWRHI      1#     42
.SWRLO      1#     54#     55
.SCATC      1#    163
.SCMTA      1#    222
.SEOP       1#   7818
.SERRO      1#  10516
.SERRT      1#  10559
.SPOWE      1#  10730
.SRDOC      1#  10462
.SREAD      1#  10359
.SSCOP      1#  10170
.STRAP      1#  10694
.STYPO      1#  10244
.STYPE      1#  10312
.STYPO      1#  10616
```

E11

DZRPSD.P11 CROSS REFERENCE TABLE

ADD 1998 2346 2441 2934 2942 3080 3148 3216 3576 3627 3687 3737 3818 3864 3961
4007 4091 4137 4223 4268 4339 4494 4508 4512 4570 6720 7401 7543 7653 7759
7913 7974 8043 8134 8170 8428 8435 8442 10277 10344 10500 10582 10645 10655
ASL 5987 10493 10495 10497 10579 10580 10581
ASLB 10282
ASR 8859 8868 8876 8884
BCC 1399 3263 5793 5812 5857 6270 6289 6348 9054 9488 9501 9518 10283
BCS 5670 5689 5738 5799 5819 5864 5988 6276 6296 6355
BEQ 1208 1301 1354 1358 1403 1405 1443 1505 1598 1610 1618 1718 1739 1923 1925
1938 1940 1948 1974 1989 2080 2095 2105 2117 2128 2138 2148 2158 2168 2178
2188 2197 2207 2218 2228 2240 2251 2262 2472 2481 2493 2501 2528 2543 2551
2559 2595 2608 2628 2649 2660 2683 2703 2714 2716 2724 2762 2771 2799 2816
2842 2850 2875 2883 2950 2961 2976 3267 3286 3297 3308 3322 3331 3338 3348
3358 3368 3378 3388 3398 3408 3417 3428 3437 3448 3458 3468 3527 4420 4429
4485 4499 4667 4777 4899 5014 5124 5251 5377 5498 5512 5603 5626 5706 5708
5837 5839 5950 5959 5969 5979 6007 6017 6024 6116 6139 6208 6219 6226 6317
6331 6337 6469 6518 6597 6614 6621 7257 7271 7280 7417 7431 7794 7803 7807
7844 7986 7988 8003 8056 8094 8116 8311 8325 8424 8431 8438 8619 8621 8625
8636 8643 8646 8656 8727 8831 8852 8919 8926 8933 8939 9005 9152 9291 9407
9413 9424 9429 9431 9450 9452 9493 9495 9505 9510 9512 9522 9562 9572 9577
9587 9644 9652 9886 9896 9910 9921 9943 9956 9969 9980 9992 10063 10136 10212
10214 10216 10220 10229 10421 10488 10535 10538 10552 10555 10584 10589 10595 10606 10672
BGE 10232
BGT 7835 10291 10490 10679
BHI 10218
BHIS 2693 9002
BIC 1215 1727 1748 1928 2216 2413 2626 2969 3050 3405 3426 3547 3594 3595 3658
3704 3705 3777 3834 3835 3895 3932 3977 3978 3979 4038 4064 4107 4108 4171
4196 4239 4240 4407 4412 4415 4450 4452 4479 4481 4514 4517 4528 4532 5510
5624 5957 5977 6013 6137 6612 6691 6694 7269 7278 7367 7428 7469 7575 7686
7832 7995 8000 8012 8114 8323 8448 8453 8458 8845 8898 8904 8913 8931 9015
9023 9097 9241 9370 9373 9376 9742 10054 10056 10082 10084 10099 10101 10109 10391
10499 10669
BIS 1588 1606 1713 1715 1726 1737 1749 2059 2070 2084 2113 2314 2317 2376 2396
2401 2414 2469 2499 2525 2540 2592 2680 2700 2735 2790 2795 2801 2806 2807
2811 2958 3035 3046 3052 3113 3181 3245 3270 3276 3318 3541 3545 3546 3587
3596 3652 3656 3657 3696 3706 3772 3775 3776 3827 3836 3894 3927 3930 3931
3970 3980 4037 4050 4062 4063 4100 4109 4170 4182 4194 4195 4232 4241 4306
4307 4311 4410 4411 4414 4417 4449 4451 4478 4480 4513 4516 4527 4531 4537
4538 4540 4541 5008 5010 5703 5834 5944 6088 6187 6199 6215 6314 6659 6669
6689 6690 6692 6693 7366 7368 7374 7375 7468 7510 7513 7517 7518 7574 7621
7627 7628 7685 7727 7733 7734 8007 8013 8450 8455 8460 8681 8689 8719 8837
8856 8865 8873 8881 8896 8897 8899 8902 8903 8905 9014 9022 9098 9232 9368
9369 9375 9489 9491 9508 9575 9734 9883 9894 10053 10055 10060 10078 10079 10081
10083 10098 10100 10108 10285 10286 10674 10675
BISB 10571
BIT 1242 1316 1357 1504 1518 1922 1971 2653 2713 2732 2939 2998 3781 5013 5707
5838 5989 6318 7985 8055 8096 8102 8118 8123 8144 8151 8158 8163 8189 8194
8423 8430 8437 8923 8925 9006 9151 9388 9451 9454 9494 9511 9561 9576 9885
9895 10211 10219 10226 10537 10544 10551
BLOS 10439
BLT 10274 10290 10352 10492 10680
BMI 8447 8452 8457 10198 10281
BNE 1138 1160 1236 1243 1317 1348 1391 1436 1519 1729 1751 1916 1972 1997 2583

```
            2590  2603  2613  2631  2654  2733  2933  2938  2940  3000  3110  3171  3178  3780  3782
            3909  4058  4190  4454  4458  4483  4488  4520  4530  4613  4639  4672  4724  4753  4782
            4844  4875  4906  4968  4994  5066  5073  5134  5149  5193  5200  5261  5276  5319  5326
            5387  5402  5491  5503  5514  5585  5608  5628  5990  5995  6121  6141  6319  6411  6418
            6479  6494  6591  6602  6616  7250  7262  7273  7422  7433  7486  7592  7698  7846  7848
            7950  7981  8006  8010  8047  8097  8103  8119  8124  8146  8153  8159  8164  8190  8192
            8195  8197  8231  8317  8327  8462  8464  8495  8501  8568  8570  8924  8993  8995  9007
            9060  9062  9100  9238  9289  9296  9378  9380  9385  9389  9405  9421  9446  9455  9526
            9528  9591  9593  9637  9650  9656  9662  9669  9739  9834  9904  9906  9929  9938  9965
            9988 10086 10089 10103 10105 10107 10111 10144 10146 10152 10227 10279 10341 10348 10393
           10398 10406 10426 10443 10449 10545 10572 10598 10670 10752
BPL        10265 10295 10335 10356 10549 10668
BR          1129  1161  1166  1170  1174  1178  1182  1186  1190  1194  1198  1202  1211  1237  1246
            1250  1320  1324  1328  1332  1336  1364  1368  1372  1376  1380  1411  1418  1424  1427
            1465  1473  1480  1507  1511  1515  1521  1935  1941  1950  1982  2278  2615  2650  2935
            3006  3268  3333  3484  3529  3785  4448  4510  5505  5610  5710  5841  5991  6123  6321
            6334  6608  7264  7424  7781  7787  7808  7888  7894  7901  7905  7909  7916  7920  8008
            8101  8107  8122  8127  8150  8157  8162  8319  8426  8433  8440  8449  8454  8459  8596
            8630  8641  8651  8840  8906  8927  8937  8996  8998  9017  9056  9084  9090  9102  9108
            9114  9118  9122  9128  9134  9153  9154  9301  9371  9393  9418  9433  9438  9440  9453
            9497  9514  9566  9581  9889  9899  9926  9948  9962  9974  9985 10080 10162 10200 10206
           10209 10222 10225 10276 10293 10337 10354 10445 10501 10514 10577 10601 10608 10646 10661
           10682 10746 10767
CLC         1724  7978  8848  9568  9583  9890  9900
CLR         1130  1136  1148  1149  1302  1342  1394  1395  1460  1590  1907  1918  2052  2399  2404
            2405  2407  2417  2418  2533  2552  2581  2585  2598  2606  2645  2688  2726  2929  3049
            3053  3260  3523  3644  3708  4383  4402  4621  4622  4623  4637  4640  4650  4652  4676
            4677  4685  4722  4732  4733  4748  4749  4763  4765  4786  4787  4800  4853  4854  4870
            4871  4885  4887  4910  4911  4923  4976  4977  4978  4992  4995  5004  5006  5071  5085
            5102  5110  5198  5212  5229  5237  5324  5338  5355  5363  5452  5453  5480  5548  5549
            5551  5568  5573  5658  5659  5725  5726  5746  5852  6063  6064  6066  6083  6089  6100
            6173  6174  6176  6194  6200  6304  6305  6343  6416  6430  6447  6455  6554  6557  6558
            6574  6576  6582  6671  6766  6846  7061  7211  7212  7239  7324  7325  7351  7408  7481
            7482  7496  7498  7587  7588  7602  7604  7708  7792  7795  7829  7830  7886  8078  8249
            8250  8252  8269  8347  8348  8350  8369  8415  8416  8427  8434  8441  8498  8556  8683
            8716  8901  8988  9046  9047  9048  9057  9382  9415  9432  9443  9498  9515  9567  9582
            9657  9912  9923  9982 10147 10186 10187 10191 10195 10224 10238 10268 10271 10378 10418
           10419 10485 10486 10570 10659 10750
CLRB        1352  2408  2793  5546  6061  6171  6555  6556 10223 10297 10450 10510
CMP         1137  1239  1402  1404  1596  1609  1616  1717  1738  1937  1939  1973  1988  2079  2094
            2104  2116  2127  2137  2147  2157  2167  2177  2187  2206  2217  2227  2239  2250  2261
            2471  2479  2491  2500  2527  2542  2550  2558  2594  2607  2627  2648  2659  2682  2692
            2702  2715  2761  2770  2797  2814  2841  2849  2874  2882  2932  2960  2970  3285  3296
            3307  3321  3337  3347  3357  3367  3377  3387  3397  3416  3427  3436  3447  3457  3467
            3526  4427  4498  5497  5511  5602  5622  5625  5705  5836  5958  5968  5978  6014  6022
            6115  6135  6138  6207  6218  6225  6316  6329  6335  6517  6596  6613  6620  7256  7270
            7279  7416  7429  7806  7845  7847  8002  8005  8009  8092  8115  8310  8321  8324  8620
            8623  8633  8642  8645  8654  8726  8932  8938  9001  9003  9423  9942  9955  9968  9991
           10207 10231 10289 10392 10397 10405 10425 10438
CMPB        2196  3407  5948 10213 10217 10347 10442 10448 10489 10491
COM         7066  7134  7983  7994  7996  7999  8001  8838  8957  8866  8874  8882
DEC         1235  1347  1390  1435  1440  1728  1750  1915  1924  1947  1996  2582  2589  2602  2630
            2936  3109  3170  3177  3266  3908  4057  4189  4453  4455  4457  4482  4484  4486  4519
            4529  4612  4638  4723  4752  4843  4874  4967  4993  5065  5072  5148  5192  5199  5275
```

DZRPSD.P11 CROSS REFERENCE TABLE

```
        5318  5325  5401  5513  5627  5994  6140  6410  6417  6493  6615  7272  7432  7485  7591
        7697  7802  7833  7949  8045  8046  8191  8196  8230  8326  8461  8463  8494  8500  8994
        9059  9061  9099  9237  9377  9379  9384  9420  9445  9525  9527  9590  9592  963E  9649
        9655  9661  9668  9738  9903  9905  9928  9964  9987 10085 10088 10102 10104 10106 10110
       10143 10151 10422 10578
DECB   10351 10667 10678
EMT       62
HALT     170  3497  4397  4646  4675  4759  4785  4881  4909  5000  5107  5137  5234  5264  5360
        5390  5478  5572  5922  6087  6198  6452  6482  6580  6656  7237  7349  7492  7598  7704
        8273  8373  8540  9137 10167 10336 10550 10745 10766
INC     1346  1432  1434  1929  1931  1949  2588  2629  2646  2691  2758  2792  2838  2871  3265
        3902  6686  7796  7831  8308  8997  9000  9506  9523  9573  9588 10230 10275 10402 10404
       10424 10540 10673 10681 10751
INCB   10235 10534
IOT       63
JMP      174   176  1162  1257  1355  1359  1384  7801  7810  7854  9915 10396
JSR     1158  1279  1315  1459  1503  1540  1565  1584  1641  1665  1689  1708  1771  1797  1822
        1847  1868  1901  1952  2051  2096  2106  2118  2129  2139  2149  2159  2169  2179  2189
        2198  2208  2219  2229  2241  2252  2263  2302  2308  2323  2337  2369  2388  2395  2421
        2432  2466  2470  2490  2523  2526  2541  2584  2593  2644  2651  2678  2681  2698  2701
        2755  2789  2834  2867  2927  2944  2948  2959  2985  2996  3025  3040  3058  3071  3102
        3125  3139  3159  3193  3207  3238  3287  3298  3309  3323  3339  3349  3359  3369  3379
        3389  3399  3409  3418  3429  3438  3449  3459  3469  3494  3514  3535  3554  3567  3604
        3617  3637  3646  3664  3679  3714  3727  3756  3766  3796  3810  3842  3854  3887  3911
        3921  3938  3952  3986  3997  4030  4044  4069  4082  4114  4126  4161  4176  4201  4214
        4246  4258  4291  4300  4317  4330  4355  4357  4378  4388  4394  4548  4561  4582  4584
        4614  4625  4643  4653  4673  4681  4689  4725  4735  4756  4766  4783  4791  4795  4804
        4845  4856  4878  4888  4907  4915  4919  4927  4969  4980  4997  5012  5086  5092  5104
        5111  5135  5153  5213  5219  5231  5238  5262  5280  5339  5345  5357  5364  5388  5406
        5441  5456  5463  5467  5475  5481  5489  5533  5538  5552  5558  5569  5574  5586  5591
        5660  5672  5679  5691  5696  5704  5728  5740  5748  5801  5821  5826  5835  5866  5873
        5918  5919  5947  5967  6001  6005  6048  6053  6067  6073  6084  6090  6101  6162  6177
        6183  6195  6201  6263  6278  6298  6306  6315  6357  6364  6431  6437  6449  6456  6480
        6498  6515  6549  6561  6567  6571  6577  6583  6589  6652  6653  6675  6699  6712  6728
        6738  6740  6770  6774  6789  6799  6818  6820  6850  6854  6869  6879  6898  6900  6930
        6934  6943  6953  6973  6975  7005  7009  7018  7028  7065  7070  7085  7095  7133  7138
        7147  7157  7176  7178  7200  7215  7222  7226  7234  7240  7248  7314  7315  7328  7334
        7338  7346  7354  7360  7379  7392  7411  7442  7444  7464  7489  7501  7521  7534  7551
        7570  7595  7612  7631  7644  7661  7681  7701  7718  7737  7750  7767  7850  7976  7977
        7991  8014  8091  8143  8253  8259  8270  8276  8318  8320  8352  8358  8370  8375  8511
        8536  8537  8557  8566  8578  8604  8690  8724  8935  8941  9009  9016  9058  9093  9148
        9243  9249  9297  9302  9383  9387  9395  9397  9399  9401  9403  9411  9422  9444  9448
        9507  9524  9574  9589  9647  9653  9659  9663  9666  9744  9755  9836  9919  9939  9953
        9966  9978  9989 10142 10150 10346 10353 10395 10546
MOV     1128  1134  1135  1139  1140  1141  1142  1143  1144  1145  1146  1147  1151  1152  1156
        1157  1206  1216  1217  1226  1227  1228  1229  1231  1232  1233  1234  1238  1240  1254
        1258  1260  1273  1274  1275  1277  1278  1294  1296  1297  1298  1299  1303  1312  1340
        1341  1343  1344  1345  1387  1388  1389  1393  1396  1397  1401  1415  1421  1428  1430
        1438  1439  1444  1456  1457  1461  1469  1476  1483  1488  1489  1502  1535  1536  1538
        1539  1560  1561  1563  1564  1582  1583  1595  1599  1600  1605  1607  1608  1611  1615
        1619  1620  1636  1637  1639  1640  1660  1661  1663  1664  1684  1685  1687  1688  1707
        1709  1710  1711  1712  1714  1716  1719  1720  1734  1735  1736  1740  1741  1742  1766
        1767  1769  1770  1792  1793  1795  1796  1817  1818  1820  1821  1842  1843  1845  1846
        1863  1864  1866  1867  1898  1899  1912  1913  1914  1919  1920  1927  1930  1934  1943
```

1945	1963	1964	1965	1966	1968	1977	1990	2056	2057	2058	2060	2061	2062	2063
2064	2065	2066	2067	2068	2071	2072	2076	2077	2078	2087	2092	2093	2102	2103
2112	2114	2115	2125	2126	2135	2136	2145	2146	2155	2156	2165	2166	2175	2176
2185	2186	2194	2195	2204	2205	2213	2214	2215	2225	2226	2237	2238	2248	2249
2259	2260	2281	2299	2300	2304	2305	2345	2347	2367	2368	2374	2375	2377	2378
2379	2380	2381	2382	2383	2384	2385	2400	2402	2403	2406	2409	2412	2416	2440
2442	2464	2465	2467	2468	2473	2482	2488	2494	2498	2502	2521	2522	2524	2529
2534	2535	2539	2544	2548	2549	2554	2555	2560	2561	2562	2578	2579	2580	2586
2587	2591	2596	2599	2600	2601	2604	2605	2609	2610	2611	2625	2643	2647	2655
2657	2658	2661	2677	2679	2684	2687	2689	2694	2699	2704	2707	2708	2709	2710
2711	2712	2717	2718	2719	2725	2727	2734	2736	2737	2738	2753	2754	2756	2757
2760	2763	2764	2769	2772	2773	2788	2791	2794	2796	2800	2802	2805	2810	2812
2813	2817	2832	2833	2835	2836	2837	2840	2843	2844	2848	2851	2852	2865	2866
2868	2869	2870	2873	2876	2877	2881	2884	2885	2925	2926	2928	2930	2931	2941
2945	2946	2947	2951	2962	2968	2977	2992	3001	3022	3023	3029	3030	3031	3032
3033	3034	3036	3037	3079	3081	3099	3103	3104	3105	3106	3107	3108	3111	3112
3147	3149	3158	3160	3165	3166	3167	3168	3172	3173	3174	3175	3176	3179	3180
3215	3217	3236	3237	3242	3243	3244	3246	3247	3248	3249	3250	3251	3252	3253
3254	3259	3261	3264	3269	3271	3274	3275	3277	3283	3284	3294	3295	3305	3306
3317	3319	3320	3332	3334	3335	3345	3346	3355	3356	3365	3366	3375	3376	3385
3386	3395	3396	3404	3406	3414	3415	3423	3424	3425	3434	3435	3445	3446	3455
3456	3465	3466	3486	3512	3513	3517	3524	3525	3528	3530	3531	3575	3577	3588
3589	3597	3598	3626	3628	3639	3643	3686	3688	3697	3698	3707	3736	3738	3754
3755	3758	3762	3789	3817	3819	3828	3829	3837	3863	3865	3885	3886	3889	3900
3901	3905	3906	3907	3913	3917	3918	3960	3962	3971	3972	3981	4006	4008	4028
4029	4032	4041	4054	4055	4056	4090	4092	4101	4102	4110	4136	4138	4159	4160
4163	4173	4186	4187	4188	4222	4224	4233	4234	4242	4267	4269	4289	4290	4293
4297	4312	4338	4340	4354	4356	4376	4377	4381	4384	4385	4405	4406	4408	4409
4418	4424	4425	4426	4430	4432	4434	4446	4447	4456	4473	4475	4477	4495	4497
4500	4502	4504	4509	4526	4542	4569	4571	4581	4583	4606	4607	4609	4610	4611
4619	4624	4631	4632	4633	4634	4635	4648	4651	4666	4717	4718	4720	4721	4729
4731	4734	4741	4742	4743	4745	4747	4750	4751	4754	4761	4764	4776	4837	4838
4840	4841	4842	4850	4852	4855	4862	4863	4865	4867	4869	4872	4873	4876	4883
4886	4898	4961	4962	4964	4965	4966	4974	4979	4986	4987	4988	4989	4990	5002
5005	5007	5015	5054	5055	5061	5062	5063	5064	5068	5078	5082	5083	5084	5093
5094	5097	5098	5109	5123	5138	5139	5142	5143	5144	5145	5146	5147	5181	5182
5188	5189	5190	5191	5195	5205	5209	5210	5211	5220	5221	5224	5225	5236	5250
5265	5266	5269	5270	5271	5272	5273	5274	5307	5308	5314	5315	5316	5317	5321
5331	5335	5336	5337	5346	5347	5350	5351	5362	5376	5391	5392	5395	5396	5397
5398	5399	5400	5435	5438	5439	5448	5454	5455	5468	5469	5470	5472	5473	5474
5479	5492	5493	5494	5496	5499	5500	5509	5530	5532	5545	5550	5559	5560	5561
5564	5565	5598	5599	5600	5601	5604	5605	5623	5651	5652	5655	5656	5657	5665
5666	5667	5684	5685	5686	5702	5722	5723	5724	5733	5734	5735	5784	5785	5789
5790	5791	5794	5795	5796	5808	5809	5810	5814	5815	5816	5833	5853	5854	5855
5859	5860	5861	5916	5917	5926	5928	5929	5930	5931	5932	5933	5935	5936	5937
5939	5940	5942	5943	5945	5946	5956	5966	5970	5971	5972	5976	6002	6003	6004
6012	6045	6047	6060	6065	6074	6075	6076	6079	6080	6107	6111	6112	6113	6114
6117	6118	6136	6159	6161	6170	6175	6184	6185	6186	6190	6191	6205	6209	6210
6214	6216	6217	6220	6223	6227	6228	6260	6262	6266	6267	6268	6271	6272	6273
6285	6286	6287	6291	6292	6293	6313	6344	6345	6346	6350	6351	6352	6399	6401
6406	6407	6408	6409	6413	6423	6427	6428	6429	6438	6439	6442	6443	6454	6468
6483	6484	6487	6488	6489	6490	6491	6492	6543	6545	6547	6559	6560	6572	6573
6575	6581	6592	6593	6594	6595	6598	6599	6611	6650	6651	6657	6658	6660	6664
6666	6667	6668	6670	6685	6687	6719	6721	6737	6739	6763	6764	6781	6782	6783

```
          6784  6785  6817  6819  6843  6844  6861  6862  6863  6864  6865  6897  6899  6923  6924
          6926  6972  6974  6998  6999  7001  7058  7059  7077  7078  7079  7080  7081  7126  7127
          7129  7175  7177  7197  7198  7207  7213  7214  7227  7228  7229  7231  7232  7233  7238
          7251  7252  7253  7255  7258  7259  7268  7277  7281  7282  7283  7289  7312  7313  7320
          7326  7327  7339  7340  7341  7343  7344  7345  7350  7363  7364  7369  7370  7372  7373
          7400  7402  7412  7413  7414  7415  7418  7419  7427  7441  7443  7462  7463  7467  7474
          7475  7476  7478  7480  7483  7484  7487  7494  7497  7508  7509  7514  7515  7516  7542
          7544  7568  7569  7573  7580  7581  7582  7584  7586  7589  7590  7593  7600  7603  7619
          7620  7624  7625  7626  7652  7654  7679  7680  7684  7691  7692  7693  7695  7696  7699
          7706  7709  7710  7725  7726  7730  7731  7732  7758  7760  7778  7779  7784  7790  7798
          7804  7805  7809  7836  7840  7843  7891  7897  7914  7924  7925  7942  7943  7944  7945
          7946  7947  7948  7951  7952  7953  7970  7971  7972  7973  7975  7984  7990  7993  7997
          7998  8004  8011  8036  8037  8038  8039  8040  8041  8044  8048  8049  8050  8057  8071
          8072  8073  8074  8076  8077  8089  8098  8104  8108  8113  8120  8125  8128  8141  8147
          8154  8160  8165  8180  8181  8182  8184  8185  8186  8187  8188  8193  8198  8223  8224
          8225  8226  8227  8228  8229  8232  8233  8234  8246  8251  8261  8262  8265  8266  8274
          8296  8297  8298  8299  8300  8301  8302  8303  8304  8305  8306  8307  8309  8313  8314
          8322  8329  8330  8331  8332  8333  8344  8349  8361  8362  8365  8366  8374  8409  8410
          8411  8412  8413  8414  8417  8418  8419  8420  8425  8432  8439  8465  8466  8467  8468
          8469  8470  8487  8488  8489  8490  8491  8492  8493  8496  8505  8508  8509  8510  8514
          8515  8516  8534  8541  8542  8543  8548  8549  8551  8552  8554  8622  8626  8627  8632
          8637  8638  8644  8647  8648  8652  8657  8658  8674  8675  8676  8678  8679  8680  8682
          8684  8688  8692  8693  8711  8712  8714  8715  8717  8718  8725  8728  8729  8733  8822
          8823  8824  8825  8826  8827  8828  8829  8836  8839  8844  8846  8855  8858  8864  8867
          8872  8875  8880  8883  8914  8915  8930  8962  8963  8964  8965  8966  8967  8982  8984
          8985  8986  8987  9026  9040  9041  9042  9043  9044  9045  9049  9050  9051  9052  9055
          9063  9064  9065  9066  9067  9068  9069  9070  9087  9095  9096  9105  9112  9125  9131
          9149  9223  9225  9226  9227  9228  9229  9230  9231  9235  9240  9242  9245  9246  9247
          9248  9306  9307  9308  9309  9310  9311  9359  9360  9361  9362  9363  9364  9365  9366
          9367  9372  9374  9381  9386  9390  9391  9392  9394  9396  9398  9400  9402  9408  9409
          9410  9416  9417  9419  9434  9435  9436  9437  9439  9442  9447  9456  9457  9458  9460
          9484  9485  9486  9490  9496  9499  9502  9503  9513  9516  9519  9520  9529  9552  9553
          9554  9555  9556  9557  9559  9560  9563  9570  9578  9585  9595  9597  9598  9599  9600
          9623  9624  9625  9627  9628  9629  9630  9631  9633  9634  9635  9638  9639  9641  9642
          9645  9646  9648  9654  9658  9660  9664  9665  9667  9671  9672  9673  9674  9675  9725
          9727  9728  9729  9730  9731  9732  9733  9736  9741  9743  9747  9748  9749  9750  9751
          9752  9753  9841  9842  9843  9844  9845  9846  9873  9874  9875  9876  9877  9878  9879
          9880  9881  9882  9884  9892  9893  9902  9908  9911  9913  9914  9917  9918  9920  9924
          9925  9941  9944  9945  9946  9947  9950  9951  9952  9954  9959  9960  9961  9967  9970
          9971  9972  9973  9976  9977  9979  9983  9984  9990  9993  9994  9995  9996  9999  10043
         10044 10045 10046 10047 10048 10049 10050 10051 10052 10061 10076 10077 10087 10095 10096
         10097 10114 10115 10116 10117 10118 10119 10132 10133 10134 10137 10138 10140 10141 10148
         10149 10154 10166 10202 10203 10205 10208 10221 10233 10234 10236 10237 10240 10241 10258
         10259 10260 10261 10262 10263 10264 10269 10272 10292 10298 10299 10300 10301 10302 10304
         10305 10338 10339 10343 10349 10379 10380 10381 10382 10384 10407 10416 10417 10427 10436
         10437 10452 10453 10454 10455 10477 10478 10479 10480 10481 10483 10484 10503 10504 10505
         10506 10507 10532 10536 10541 10553 10556 10569 10574 10583 10588 10593 10594 10596 10599
         10603 10610 10611 10642 10650 10651 10652 10658 10665 10683 10684 10685 10686 10687 10703
         10704 10707 10735 10736 10737 10738 10739 10740 10741 10742 10743 10744 10749 10753 10754
         10755 10756 10757 10758 10759 10760 10763
MOVB      1150  1350  1462  1932  2331  2429  2759  2768  2809  2839  2847  2872  2880  3066  3133
          3201  3562  3613  3673  3722  3804  3851  3946  3994  4077  4122  4209  4254  4325  4556
          5080  5081  5095  5096  5140  5141  5207  5208  5222  5223  5267  5268  5333  5334  5348
          5349  5393  5394  5450  5451  5471  5547  5562  5563  6062  6077  6078  6172  6188  6189
```

```
        6425  6426  6440  6441  6485  6486  6707  7209  7210  7230  7322  7323  7342  7387  7529
        7639  7745  8247  8248  8263  8264  8345  8346  8363  8364  8506  8507  8544  8546 10239
       10267 10270 10284 10287 10296 10340 10357 10390 10403 10423 10441 10446 10487 10543 10643
       10644 10647 10648 10649 10653 10656 10657 10676 10706
MEG    10266 10654
NOP     3008  4413  4416  4515  5009  5764  5891  6382  6809  6889  6963  7038  7105  7167  7851
        7852  7853  8015  8720  8721  8722  8723  9245  9258  9266  9277  9747  9764  9774  9787
        9799  9807  9820
RESET   1314  2394  7849  7989
ROL     1725  1747  5669  5688  5737  5792  5798  5811  5818  5856  5863  6269  6275  6288  6295
        6347  6354  7979 10197 10494 10496 10498 10660 10662 10663 10664 10666
ROR     1398  3262  8421  8422  8444  8445  8849  8850  8894  8895  8911  8912  9053  9487  9500
        9517  9565  9569  9580  9584  9638  9891  9898  9901
RTI     7926  8058  8200 10242 10306 10345 10401 10408 10428 10456 10508 10558 10688 10765
RTS     1283  1544  1569  1645  1669  1693  1775  1801  1826  1851  1872  2286  2352  2447  3084
        3152  3220  3491  3582  3633  3692  3741  3824  3869  3966  4013  4096  4143  4228  4274
        4344  4573  4697  4703  4812  4822  4935  4945  5161  5165  5288  5292  5414  5418  5756
        5762  5883  5889  6374  6380  6506  6510  6723  7405  7547  7657  7763  7954  7982  8016
        8051  8079  8132  8135  8168  8171  8235  8279  8334  8378  8471  8517  8586  8594  8612
        8617  8662  8694  8734  8968  9027  9071  9312  9461  9530  9601  9677  9848 10001 10120
       10155 10358 10385 10613 10708
SEC     1746  5668  5687  5736  5797  5817  5862  6274  6294  6353  9564  9579  9887  9897
SUB     1944  4431  4433  4501  4503  5501  5606  6119  6670  7260  7420  8042  8090  8142  8183
        8315  8535  8713  8983  9224  9414  9640  9726  9922  9958  9981 10273 10542
TRAP   10710 10720 10721 10722 10723 10724 10725 10726 10727 10728 10729
TST     1159  1207  1442  1486  1487  1921  2269  2274  2285  2612  2623  2632  2722  2949  3330
        3475  3480  3490  3493  3779  4419  4671  4781  4905  5067  5133  5194  5260  5320  5386
        5490  5502  5508  5584  5607  5993  6120  6412  6478  6590  6601  6610  7249  7261  7267
        7421  7426  7793  7980  8316  8446  8451  8456  8567  8569  8618  8830  8851  8916  8992
        9150  9288  9290  9292  9404  9406  9412  9425  9430  9449  9492  9504  9509  9521  9571
        9586  9643  9651  9833  9909  9937 10062 10135 10145 10204 10228 10278 10288 10342 10383
       10400 10420 10502 10509 10548 10554 10605 10671 10705
TSTB    1300  1353  6006  7987  8545  8547 10215 10280 10294 10334 10355 10597
.ASCII   275   276 10829 10834 10843 10851 10858 10868 10898 10913 10928 10937 10944 10978 10988
       10997 11024 11042 11056 11062 11064 11073 11087 11091 11106 11120 11140 11168 11186 11206
       11226 11241 11253 11273 11296 11314 11332 11343 11354 11380 11403 11423 11446 11463 11489
.ASCIZ   274   277  1168  1172  1176  1180  1184  1188  1192  1196  1200  1204  1213  1248  1252
        1322  1326  1330  1334  1338  1366  1370  1374  1378  1382  1413  1420  1426  1467  1475
        1482  1509  1513  1517  1523  3787  7783  7789  7855  7890  7896  7903  7907  7911  7918
        7922  9086  9092  9104  9110  9116  9120  9124  9130  9136 10164 10460 10614 10769 10781
       10790 10794 10800 10805 10808 10815 10818 10822 10825 10876 10882 10892 10909 10918 10952
       10960 10968 11001 11011 11032 11050 11082 11088 11099 11113 11130 11152 11163 11177 11196
       11216 11234 11247 11263 11285 11305 11323 11367 11392 11413 11435 11455 11476 11493
.BLKB  10366 10459
.BLKW   1041  1109  1110  9700  9701  9702  9703  9704  9705  9707  9708  9709  9710  9711 10311
.BYTE    236   237   242   243   254   255   256   257  1117  1979  1980  1992  1993  7858 10457
       10458 10689 10690 10691 10692 11568 11570 11572 11575 11578 11581 11582 11584 11588 11591
       11593 11595 11598 11601 11604 11606 11609 11612 11615 11618 11621 11623 11627
.ENABL     1
.END   11630
.ENDC      6    17    23    33    40    50    52    53    54    62   148   162   175   196   207
         218   220   223   231   233   258   266   272   273   274   275   278   279   724   726
         764   766  1134  1139  1140  1142  1144  1146  1148  1149  1151  1153  1168  1172  1176
        1180  1184  1188  1192  1196  1200  1204  1213  1222  1223  1224  1225  1226  1227  1248
```

```
        1252  1266  1267  1271  1272  1273  1274  1291  1292  1293  1294  1295  1309  1310  1311
        1312  1313  1322  1326  1330  1334  1338  1366  1370  1374  1378  1382  1413  1420  1426
        1449  1450  1454  1455  1456  1457  1458  1467  1475  1482  1494  1495  1500  1501  1502
        1509  1513  1517  1523  1528  1529  1532  1533  1534  1553  1554  1557  1558  1559  1575
        1576  1579  1580  1581  1629  1630  1633  1634  1635  1653  1654  1657  1658  1659  1677
        1678  1681  1682  1683  1700  1701  1704  1705  1706  1758  1759  1762  1763  1764  1784
        1785  1788  1789  1790  1810  1811  1814  1815  1816  1835  1836  1839  1840  1841  1856
        1857  1860  1861  1862  1880  1886  1890  1891  1896  1897  1898  1980  1981  1993  1994
        2009  2018  2021  2033  2039  2040  2048  2049  2050  2095  2105  2128  2138  2148  2158
        2168  2178  2188  2207  2251  2262  2290  2291  2297  2298  2299  2359  2360  2365  2366
        2367  2454  2455  2462  2463  2464  2507  2508  2519  2520  2521  2570  2571  2576  2577
        2578  2637  2638  2641  2642  2643  2668  2669  2674  2675  2676  2746  2747  2750  2751
        2752  2781  2782  2785  2786  2787  2825  2826  2829  2830  2831  2858  2859  2862  2863
        2864  2892  2902  2909  2913  2917  2918  2922  2923  2924  3013  3014  3020  3021  3022
        3031  3092  3097  3098  3099  3226  3227  3234  3235  3236  3286  3297  3308  3348  3358
        3368  3378  3388  3398  3417  3458  3468  3503  3504  3510  3511  3512  3746  3747  3752
        3753  3754  3787  3876  3877  3883  3884  3[illegible]5  4019  4020  4026  4027  4028  4150  4151
        4157  4158  4159  4280  4281  4287  4288  [illegible]289  4351  4352  4353  4354  4365  4366  4374
        4375  4376  4578  4579  4580  4581  4596  4597  4604  4605  4606  4707  4708  4715  4716
        4717  4828  4829  4835  4836  4837  4952  4953  4959  4960  4961  5021  5041  5046  5047
        5052  5053  5054  5140  5160  5173  5174  5179  5180  5181  5267  5287  5299  5300  5305
        5306  5307  5393  5413  5425  5426  5432  5433  5434  5492  5513  5521  5522  5528  5529
        5530  5636  5637  5643  5644  5645  5767  5768  5775  5776  5777  5901  5902  5913  5914
        5915  6035  6036  6043  6044  6045  6149  6150  6157  6158  6159  6241  6242  6251  6252
        6253  6388  6389  6395  6396  6397  6485  6505  6531  6532  6538  6539  6540  6631  6632
        6647  6648  6649  6734  6735  6736  6737  6748  6749  6759  6760  6761  6771  6809  6814
        6815  6816  6817  6828  6829  6839  6840  6841  6851  6889  6894  6895  6896  6897  6908
        6909  6919  6920  6921  6931  6963  6969  6970  6971  6972  6983  6984  6994  6995  6996
        7006  7038  7043  7044  7054  7055  7056  7067  7105  7111  7112  7122  7123  7124  7135
        7167  7172  7173  7174  7175  7187  7188  7194  7195  7196  7250  7251  7271  7272  7298
        7299  7310  7311  7312  7438  7439  7440  7441  7452  7453  7460  7461  7462  7555  7556
        7566  7567  7568  7666  7667  7677  7678  7679  7769  7770  7776  7777  7778  7779  7783
        7789  7819  7822  7823  7824  7826  7829  7835  7838  7839  7843  7855  7858  7859  7862
        7881  7890  7896  7903  7907  7911  7918  7922  9086  9092  9104  9110  9116  9120  9124
        9130  9136  9158  9159  9218  9221  9315  9337  9344  9346  9471  9473  9538  9540  9609
        9612  9687  9693  9719  9721  9851  9853 10011 10013 10124 10126 10164 10171 10177 10182
       10197 10199 10210 10213 10214 10215 10217 10219 10226 10230 10235 10240 10243 10244 10245
       10313 10360 10397 10410 10429 10430 10437 10439 10444 10459 10460 10461 10463 10472 10516
       10517 10523 10534 10541 10547 10548 10558 10559 10560 10578 10616 10617 10695 10704 10707
       10709 10719 10720 10721 10722 10723 10724 10725 10726 10727 10728 10729 10731 10743 10753
       10763 10765 10772

.EQUIV   62    63    65    80    81   110   111   112   113   114   115   116   117   118   119
        138   139   140   141   142   143   144   145   146   147

.EVEN   1168  1172  1176  1180  1184  1188  1192  1196  1200  1204  1213  1248  1252  1322  1326
        1330  1334  1338  1366  1370  1374  1378  1382  1413  1420  1426  1467  1475  1482  1509
        1513  1517  1523  3787  7783  7789  7890  789[illegible]  7903  7907  7911  7918  7922  9086  9092
        9104  9110  9116  9120  9124  9130  9136 10164 10368 10615 10771 11498 11628

.IF        2    14    20    32    39    50    51    52    53    54    60   120   148   170   196
         207   218   219   222   228   232   258   266   272   273   274   278   723   725   763
         765  1134  1139  1140  1142  1144  1146  1148  1149  1151  1167  1171  1175  1179  1183
        1187  1191  1195  1199  1203  1212  1221  1223  1224  1226  1227  1247  1251  1265  1267
        1271  1273  1274  1290  1292  1294  1295  1308  1310  1312  1313  1321  1325  1329  1333
        1337  1365  1369  1373  1377  1381  1412  1419  1425  1448  1450  1454  1456  1457  1466
        1474  1481  1493  1495  1500  1502  1508  1512  1516  1522  1527  1529  1532  1534  1552
```

```
1554  1557  1559  1574  1576  1579  1581  1628  1630  1633  1635  1652  1654  1657  1659
1676  1678  1681  1683  1699  1701  1704  1706  1757  1759  1762  1764  1783  1785  1788
1790  1809  1811  1814  1816  1834  1836  1839  1841  1855  1857  1860  1862  1879  1885
1889  1891  1896  1898  1979  1980  1992  1993  2006  2017  2020  2030  2038  2040  2048
2050  2094  2095  2104  2105  2127  2128  2137  2138  2147  2148  2157  2158  2167  2168
2177  2178  2187  2188  2206  2207  2250  2251  2261  2262  2289  2291  2297  2299  2358
2360  2365  2367  2453  2455  2462  2464  2506  2508  2519  2521  2569  2571  2576  2578
2636  2638  2641  2643  2667  2669  2674  2676  2745  2747  2750  2752  2780  2782  2785
2787  2824  2826  2829  2831  2857  2859  2862  2864  2891  2901  2908  2912  2916  2918
2922  2924  3012  3014  3020  3022  3090  3092  3097  3099  3225  3227  3234  3236  3285
3286  3296  3297  3307  3308  3347  3348  3357  3358  3367  3368  3377  3378  3387  3388
3397  3398  3416  3417  3457  3458  3467  3468  3502  3504  3510  3512  3745  3747  3752
3754  3786  3875  3877  3883  3885  4018  4020  4026  4028  4149  4151  4157  4159  4279
4281  4287  4289  4350  4352  4354  4364  4366  4374  4376  4577  4579  4581  4595  4597
4604  4606  4706  4708  4715  4717  4827  4829  4835  4837  4951  4953  4959  4961  5020
5040  5045  5047  5052  5054  5139  5159  5172  5174  5179  5181  5266  5286  5298  5300
5305  5307  5392  5412  5424  5426  5432  5434  5491  5492  5512  5513  5520  5522  5528
5530  5635  5637  5643  5645  5766  5768  5775  5777  5900  5902  5913  5915  6034  6036
6043  6045  6148  6150  6157  6159  6240  6242  6251  6253  6387  6389  6395  6397  6484
6504  6530  6532  6538  6540  6630  6632  6647  6649  6733  6735  6737  6747  6749  6759
6761  6771  6808  6813  6815  6817  6827  6829  6839  6841  6851  6888  6893  6895  6897
6907  6909  6919  6921  6931  6962  6968  6970  6972  6982  6984  6994  6996  7006  7037
7042  7044  7054  7056  7066  7104  7110  7112  7122  7124  7134  7166  7171  7173  7175
7186  7188  7194  7196  7250  7271  7297  7299  7310  7312  7437  7439  7441  7451  7453
7460  7462  7554  7556  7566  7568  7665  7667  7677  7679  7768  7770  7776  7778  7779
7782  7788  7818  7822  7823  7824  7825  7826  7828  7834  7837  7839  7843  7854  7855
7861  7880  7889  7895  7902  7906  7910  7917  7921  9085  9091  9103  9109  9115  9119
9123  9129  9135  9157  9158  9217  9220  9314  9336  9343  9345  9470  9472  9537  9539
9608  9611  9686  9692  9718  9720  9850  9852 10010 10012 10123 10125 10163 10170 10176
10181 10186 10197 10209 10211 10212 10213 10215 10216 10217 10226 10228 10237 10242 10243
10244 10312 10359 10363 10392 10409 10429 10437 10438 10443 10459 10460 10462 10468 10484
10516 10522 10532 10537 10544 10546 10547 10548 10551 10558 10559 10577 10593 10616 10694
10703 10707 10709 10710 10720 10721 10722 10723 10724 10725 10726 10727 10728 10729 10730
10743 10753 10761 10763 10769

.IFF    33    40    50    52    53    54    62   220   223   232   258   279   724   726   764
       766  1139  1222  1223  1225  1226  1227  1266  1267  1272  1273  1274  1291  1292  1293
      1294  1295  1309  13.0  1311  1312  1313  1449  1450  1455  1456  1457  1494  1495  1501
      1502  1528  1529  1533  1534  1553  1554  1558  1559  1575  1576  1580  1581  1629  1630
      1634  1635  1653  1654  1658  1659  1677  1678  1682  1683  1700  1701  1705  1706  1758
      1759  1763  1764  1784  1785  1789  1790  1810  1811  1815  1816  1835  1836  1840  1841
      1856  1857  1861  1862  1880  1886  1890  1891  1897  1898  1979  1980  1992  1993  2018
      2021  2039  2040  2049  2050  2290  2291  2298  2299  2359  2360  2366  2367  2454  2455
      2463  2464  2507  2508  2520  2521  2570  2571  2577  2578  2637  2638  2642  2643  2668
      2669  2675  2676  2746  2747  2751  2752  2781  2782  2786  2787  2825  2826  2830  2831
      2858  2859  2863  2864  2892  2902  2909  2913  2917  2918  2923  2924  3013  3014  3021
      3022  3091  3092  3098  3099  3226  3227  3235  3236  3503  3504  3511  3512  3746  3747
      3753  3754  3876  3877  3884  3885  4019  4020  4027  4028  4150  4151  4158  4159  4280
      4281  4288  4289  4351  4352  4353  4354  4365  4366  4375  4376  4578  4579  4580  4581
      4596  4597  4605  4606  4707  4708  4716  4717  4828  4829  4836  4837  4952  4953  4960
      4961  5021  5041  5046  5047  5053  5054  5140  5160  5173  5174  5180  5181  5267  5287
      5299  5300  5306  5307  5393  5413  5425  5426  5433  5434  5521  5522  5529  5530  5636
      5637  5644  5645  5767  5768  5776  5777  5901  5902  5914  5915  6035  6036  6044  6045
      6149  6150  6158  6159  6241  6242  6252  6253  6388  6389  6396  6397  6484  6504  6531
      6532  6539  6540  6631  6632  6648  6649  6734  6735  6736  6737  6748  6749  6760  6761
```

```
        6808  6814  6815  6816  6817  6828  6829  6840  6841  6888  6894  6895  6896  6897  6908
        6909  6920  6921  6962  6969  6970  6971  6972  6983  6984  6995  6996  7037  7043  7044
        7055  7056  7105  7111  7112  7123  7124  7167  7172  7173  7174  7175  7187  7188  7195
        7196  7298  7299  7311  7312  7438  7439  7440  7441  7452  7453  7461  7462  7555  7556
        7567  7568  7666  7667  7678  7679  7769  7770  7777  7778  7779  7819  7825  7829  7835
        7838  7855  7862  7881  9158  9159  9218  9221  9315  9337  9344  9346  9471  9473  9538
        9540  9609  9612  9687  9693  9719  9721  9851  9853 10011 10013 10124 10126 10171 10210
       10213 10214 10217 10243 10245 10313 10360 10410 10429 10430 10438 10443 10459 10463 10517
       10522 10537 10558 10559 10560 10577 10593 10617 10695 10704 10731 10763
.IFT    1168  1172  1176  1180  1184  1188  1192  1196  1200  1204  1213  1248  1252  1322  1326
        1330  1334  1338  1366  1370  1374  1378  1382  1413  1420  1426  1467  1475  1482  1509
        1513  1517  1523  3787  5139  5159  5266  5286  5392  5412  6484  6504  6771  6808  6851
        6888  6931  6962  7006  7037  7066  7104  7134  7166  7783  7789  7890  7896  7903  7907
        7911  7918  7922  9086  9092  9104  9110  9116  9120  9124  9130  9136 10164 10225 10416
       10489 10509 10516 10547
.IFTF   1168  1172  1176  1180  1184  1188  1192  1196  1200  1204  1213  1248  1252  1322  1326
        1330  1334  1338  1366  1370  1374  1378  1382  1413  1420  1426  1467  1475  1482  1509
        1513  1517  1523  3787  7783  7789  7890  7896  7903  7907  7911  7918  7922  9086  9092
        9104  9110  9116  9120  9124  9130  9136 10164 10223 10410 10485 10493 10515 10546
.IIF       1     6    11    47    48    49    50    53    54    55    56   170   278  1140  1142
        1148  1149  1151  1152  1477  1484  1978  1991  7823  7829  7830  7841  7855  7859 10177
       10178 10179 10180 10181 10182 10224 10225 10240 10243 10244 10359 10363 10368 10452 10460
       10461 10516 10523 10524 10525 10526 10527 10559 10575 10600 10604 10719 10720 10721 10722
       10723 10724 10725 10726 10727 10728 10729
.IRP    1134  1221  1265  1290  1308  1448  1493  1527  1552  1574  1628  1652  1676  1699  1757
        1783  1809  1834  1855  1889  2038  2289  2358  2453  2506  2569  2636  2667  2745  2780
        2824  2857  2916  3012  3090  3225  3502  3745  3875  4018  4149  4279  4350  4364  4577
        4595  4706  4827  4951  5045  5172  5298  5424  5520  5635  5766  5900  6034  6148  6240
        6387  6530  6630  6733  6747  6813  6827  6893  6907  6968  6982  7042  7110  7171  7186
        7297  7437  7451  7554  7665  7768  7942  7951  8036  8048  8223  8232  8296  8329  8409
        8411  8466  8487  8514  8674  8692  8711  8733  8822  8962  8985  9026  9040  9065  9225
        9306  9364  9460  9484  9529  9552  9597  9627  9671  9727  9841  9878  9999 10044 10114
       10134 10154 10186 10258 10298 10479 10505 10532 10737 10753
.LIST      1    13    24    41    53   162   170   258   260   261   262   263   264   265   266
         267   268   269   270   271   272  1134  1154  1168  1172  1176  1180  1184  1188  1192
        1196  1200  1204  1213  1221  1226  1228  1248  1252  1265  1273  1275  1290  1294  1296
        1308  1312  1322  1326  1330  1334  1338  1366  1370  1374  1378  1382  1413  1420  1426
        1448  1456  1467  1475  1482  1493  1502  1509  1513  1517  1523  1527  1534  1536  1552
        1559  1561  1574  1581  1583  1628  1635  1637  1652  1659  1661  1676  1683  1685  1699
        1706  1707  1757  1764  1767  1783  1790  1793  1809  1816  1818  1834  1841  1843  1855
        1862  1864  1889  1898  1899  2005  2034  2038  2050  2289  2299  2300  2358  2367  2368
        2453  2464  2465  2506  2521  2522  2563  2578  2636  2643  2667  2676  2677  2745  2752
        2753  2780  2787  2788  2824  2831  2832  2857  2864  2866  2916  2924  2926  3012  3022
        3023  3090  3099  3225  3236  3237  3502  3512  3513  3745  3754  3755  3787  3875  3885
        3886  4018  4028  4029  4149  4159  4160  4279  4289  4290  4350  4354  4364  4376  4377
        4577  4581  4595  4606  4607  4706  4717  4718  4827  4837  4838  4951  4961  4962  5045
        5054  5055  5172  5181  5182  5298  5307  5308  5424  5434  5435  5439  5520  5530  5532
        5635  5645  5652  5766  5777  5785  5900  5915  5917  6034  6045  6047  6148  6159  6161
        6240  6253  6262  6387  6397  6401  6530  6540  6545  6630  6649  6651  6733  6737  6747
        6761  6764  6813  6817  6827  6841  6844  6893  6897  6907  6921  6924  6968  6972  6982
        6996  6999  7042  7056  7059  7110  7124  7127  7171  7175  7186  7196  7198  7297  7312
        7313  7437  7441  7451  7462  7463  7554  7568  7569  7665  7679  7680  7768  7778  7783
        7789  7890  7896  7903  7907  7911  7918  7922  9086  9092  9104  9110  9116  9120  9124
        9130  9136 10164 10181 10429 10709 10710 10719 10720 10721 10722 10723 10724 10725 10726
```

N11


```
            10727   10728   10729   10730
.MACRO         41      54     222    1221    1264    1447    1493    1527    1551    1573    1627    1651    1675    1698    1756
             1782    1808    1833    1854    1[illegible]8    2036    2268    2357    2452    2505    2568    2635    2666    2744    2779
             2823    2856    2915    3011    3[illegible]9    3224    3501    3744    3874    4017    4148    4278    4363    4594    4705
             4826    4950    5044    5171    5297    5423    5519    5632    5765    5898    6033    6147    6239    6386    6529
             6629    6747    6906    7185    7296    7450    7553    7664    7768   10710
.MCALL          1     162
.NLIST          1      13      24      41      53     162     170     258     260     261     262     263     264     265     266
              267     268     269     270     271     272    1134    1154    1168    1172    1176    1180    1184    1188    1192
             1196    1200    1204    1213    1221    1226    1228    1248    1252    1265    1273    1275    1290    1294    1296
             1308    1312    1322    1326    1330    1334    1338    1366    1370    1374    1378    1382    1413    1420    1426
             1448    1456    1467    1475    1482    1493    1502    1509    1513    1517    1523    1527    1534    1536    1552
             1559    1561    1574    1581    1583    1628    1635    1637    1652    1659    1661    1676    1683    1685    1699
             1706    1707    1757    1764    1767    1783    1790    1793    1809    1816    1818    1834    1841    1843    1855
             1862    1864    1889    1898    1899    2005    2034    2038    2050    2289    2299    2300    2358    2367    2368
             2453    2464    2465    2506    2521    2522    2569    2578    2636    2643    2667    2676    2677    2745    2752
             2753    2780    2787    2788    2824    2831    2832    2857    2864    2866    2916    2924    2926    3012    3022
             3023    3090    3099    3225    3236    3237    3502    3512    3513    3745    3754    3755    3787    3875    3885
             3886    4018    4028    4029    4149    4159    4160    4279    4289    4290    4350    4354    4364    4376    4377
             4577    4581    4595    4606    4607    4706    4717    4718    4827    4837    4838    4951    4961    4962    5045
             5054    5055    5172    5181    5182    5298    5307    5308    5424    5434    5435    5439    5520    5530    5532
             5635    5645    5652    5766    5777    5785    5900    5915    5917    6034    6045    6047    6148    6159    6161
             6240    6253    6262    6387    6397    6401    6530    6540    6545    6630    6649    6651    6733    6737    6747
             6761    6764    6813    6817    6827    6841    6844    6893    6897    6907    6921    6924    6968    6972    6982
             6996    6999    7042    7056    7059    7110    7124    7127    7171    7175    7186    7196    7198    7297    7312
             7313    7437    7441    7451    7462    7463    7554    7568    7569    7665    7679    7680    7768    7778    7783
             7789    7890    7896    7903    7907    7911    7918    7922    9086    9092    9104    9110    9116    9120    9124
             9130    9136   10164   10181   10429   10709   10710   10719   10720   10721   10722   10723   10724   10725   10726
            10727   10728   10729   10730
.PAGE          57     163     183     222     278    8052    9211   10244   10312   10359   10462   10516   10559   10616   10694
            10730
.REPT          14      20     170     260     266    2006    2030
.SBTTL         41      43      58     164     171     184     224     280     964    1122    1123    1124    1126    1221    1265
             1290    1308    1448    1493    1525    1527    1552    1574    1628    1652    1676    1699    1757    1783    1809
             1834    1855    1889    2038    2289    2358    2450    2453    2506    2569    2636    2667    2742    2745    2780
             2824    2857    2889    2916    3012    3090    3225    3502    3745    3875    4018    4149    4279    4350    4364
             4577    4592    4595    4706    4827    4951    5045    5172    5298    5424    5520    5635    5766    5896    5900
             6034    6148    6240    6387    6530    6630    6733    6747    6813    6827    6893    6907    6968    6982    7042
             7110    7171    7186    7297    7437    7451    7554    7665    7768    7813    7814    7815    7820    7930    7960
             8019    8052    8069    8082    8212    8240    8284    8380    8698    8737    9077    9156   10168   10172   10246
            10314   10361   10464   10518   10561   10618   10696   10711   10732
.TITLE          1
.WORD         170     235     238     239     240     241     244     245     246     247     248     249     258     260     261
              262     263     264     265     266     267     268     269     270     271    1042    1043    1045    1047    1048
             1104    1280    1281    1541    1542    1566    1567    1642    1643    1666    1667    1690    1691    1772    1773
             1798    1799    1823    1824    1848    1849    1869    1870    2537    2696    2994    3115    3118    3183    3186
             4684    4794    4918    4922    5442    5443    5444    5534    5535    5536    5539    5540    5541    5661    5662
             5663    5673    5674    5675    5680    5681    5682    5692    5693    5694    5729    5730    5731    5741    5742
             5743    5802    5803    5804    5822    5823    5824    5867    5868    5869    6049    6050    6051    6054    6055
             6056    6163    6164    6165    6279    6280    6281    6299    6300    6301    6358    6359    6360    6550    6551
             6552    7201    7202    7203    7316    7317    7318    7834    7837    8691    9222    9244    9251    9252    9253
             9254    9255    9299    9300    9303    9304    9724    9745    9837    9838   10363   10364   10365   10512   10515
            10586   10591   10693   10762   10764   11500   11502   11504   11507   11510   11513   11515   11518   11522   11525
            11527   11528   11531   11534   11538   11541   11544   11548   11551   11554   11557   11560   11565
```

ERRORS DETECTED: 0

*DZRPSD,DZRPSD/SOL/CRF=DZRPSD
RUN-TIME: 98 110 16 SECONDS
CORE USED: 23K

Spooler runtime 13 Seconds, 81 KCS, 1255 disk reads, 0 disk writes, 350 pages

