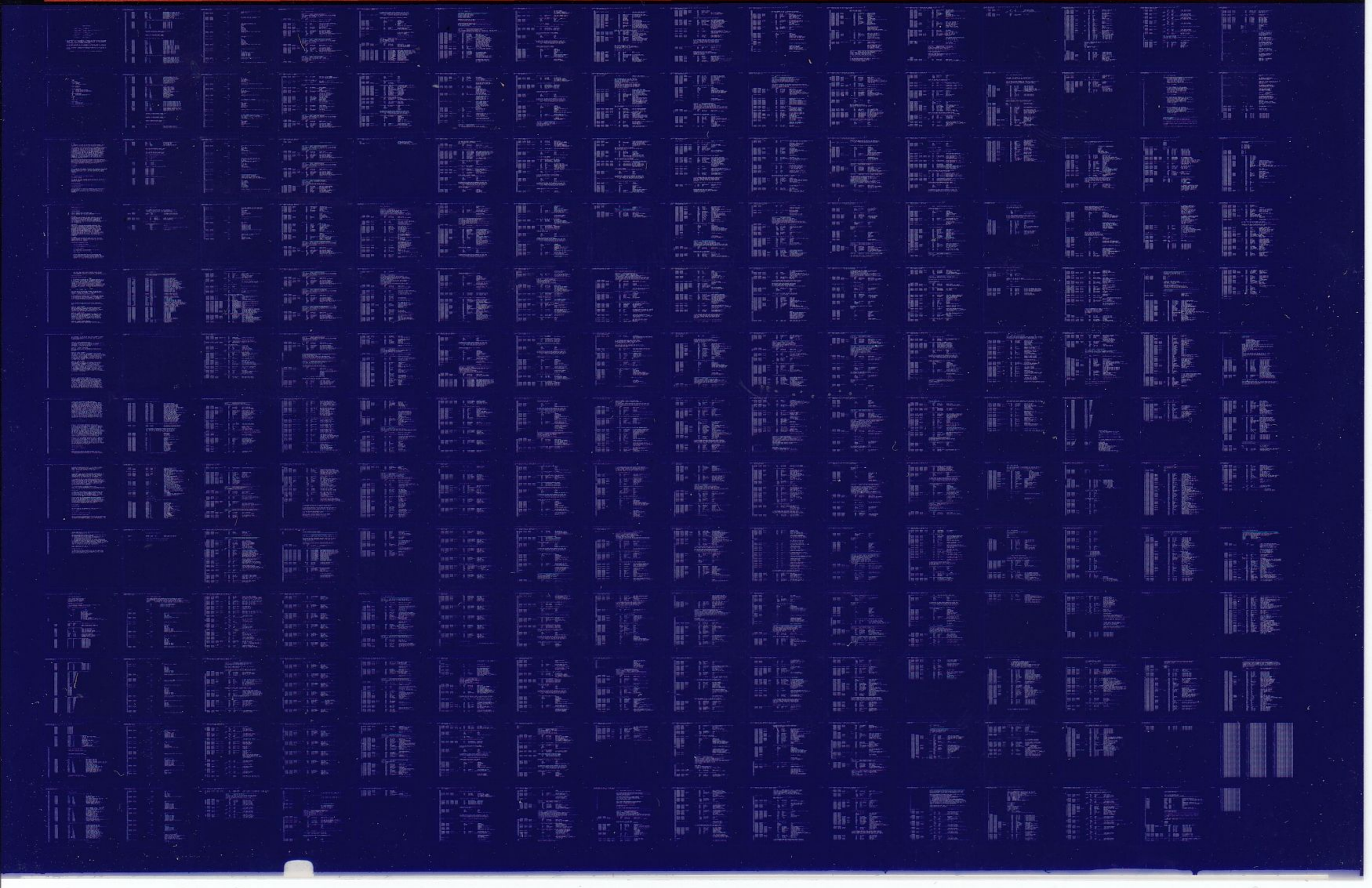
RP04
RP05, RP06

RP04/5/6 DSKLS 1
CZRJGEO

AH-9210E-MC
FICHE 1 OF 2



RP04
RP05,RP06

RP04/5/6 DSKLS 1
CZRJGE0

AH-9210E-MC
FICHE 2 OF 2

```
                        .REM    @
```

IDENTIFICATION

PRODUCT CODE: AC-9208E-MC

PRODUCT NAME: CZRJGE0 RP04/5/6 DISKLESS TEST, PT 1

PRODUCT DATE: NOVEMBER 1981

MAINTAINER: CX DIAGNOSTIC GROUP

AUTHOR: PETE BLACKSTONE

REVISED BY: MIKE LEAVITT

CONTENTS

1.0 ABSTRACT

THE DIAGNOSTIC IS USED TO TEST RP04/5/6 DEVICE CONTROL LOGIC
CONNECTED TO EITHER AN RH11 OR RH70 DISK DRIVE CONTROLLER

THIS DIAGNOSTIC TESTS THE RH11 AND DCL OF AN RJP04/5/6 SUBSYSTEM.
IT DOES NOT USE THE DISK SURFACE OR ANY SIGNALS FROM THE MDLI.
IT REQUIRES THAT THE DCL CABLE BE PLUGGED INTO THE MDLI OR
BE APPROPIATELY TERMINATED. IF THE DISK IS POWERED UP, IT
IS REQUIRED TO GET THE DISK TO THE "HEADS UNLOADED" POSITION.
AFTER A SUCCESSFUL RUN (WITH NO ERRORS) OF THIS DIAGNOSTIC
IT CAN BE ASSERTED THAT, "THAT PART OF THE DCL THAT HANDLES
DATA OR DATA ASSOCIATED LOGIC IS WORKING PROPERLY". THIS
IMPLIES THAT, THE 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.

THE DIAGNOSTIC DOES NOT DO ANY TESTING OF THE RH70 CONTROLLER
WHEN IT IS USED ON AN RWP04/5/6 SYSTEM TO TEST RP04/5/6 DISK DRIVES
CONNECTED TO THAT TYPE OF CONTROLLER. IT IS ASSUMED THAT THE
RH70 SPECIFIC CONTROLLER DIAGNOSTIC HAVE BEEN SUCCESSFULLY
RUN TO COMPLETION BEFORE THIS DIAGNOSTIC IS RUN.

2.0 REQUIREMENTS

2.1 EQUIPMENT

PDP-11 COMPUTER WITH CONSOLE TELETYPE, AND A RP04/5/6 DISK SYSTEM.
THE RP04/5/6 DISK SYSTEM WILL CONSIST OF AN RH11/RH70 CONTROLLER,
AND DISK CONTROL LOGIC (DCL), THE CABLE FROM THE DCL CAN BE
CONNECTED TO THE MDLI, BUT IF NOT THAT CABLE MUST BE PROPERLY
TERMINATED.

2.2 STORAGE

THIS PROGRAM REQUIRES 16K WORDS OF MEMORY.

2.3 PRELIMINARY PROGRAMS

THIS CAN BE THE FIRST PROGRAM RUN ON AN RJP04/5/6 SYSTEM BUT THE
CONTROLLER DIAGNOSTICS MUST BE RUN FIRST IN THE CASE OF AN
RWP04/5/6 SYSTEM.

3.0 LOADING PROCEDURE

USE STANDARD PROCEDURE FOR LOADING .ABS TAPES

4.0 STARTING PROCEDURE

SWITCH 12 MUST BE SET WHEN THIS PROGRAM IS TO BE RUN USING AN
RH70 CONTROLLER. IT CAN BE SET AT THE FRON PAEL, OR IN THE
SOFTWARE SWITCH REGISTER IF THE OPERATOR SO DESIRES. SE
PARAGRAPH 5.1 FOR A DESCRIPTION OF SOFTWARE SWITCH REGISTER
OPERATION.

4.1 CONTROL SWITCH SETTINGS

SEE SECTION 5.1

4.2 STARTING ADDRESS

START AT ADDRESS 200---FOR NORMAL RUN
START AT ADDRESS 204---TO SELECT NON-DEFAULT ADDRESSES
START AT ADDRESS 210---FOR UNIT SELECTION

200 START
ALL SWITCHES MUST BE DOWN FOR WORST CASE RUN. WITH THIS STARTING ADDRESS ALL THE RP04/5/6S 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.

204 RESTART
SAME AS START 200 WITH THE FOLLOWING EXCEPTION: THE PROGRAM WILL INTERROGATE THE OPERATOR FOR A NON-STANDARD C.S.R. AND VECTOR ADDRESS BEFORE STARTING. ONCE THE QUESTIONS HAVE BEEN CORRECTLY ANSWERED, AND IT IS ALSO NECESSARY TO SELECT A PARTICULAR UNIT FOR TEST (TYPICAL PROGRAM EXECUTION FROM ADDRESS 210), THE PROCESSOR MAY BE HALTED AND RESTARTED FROM ADDRESS 210. THE NEW PARAMETERS WILL NOT BE CHANGED UNLESS THE PROGRAM IS AGAIN RESTARTED FROM ADDRESS 204. IF ALL UNITS ARE TO BE CHECKED, THE PROCESSOR NEED NOT BE TOUCHED. THE PROGRAM WILL AUTOMATICALLY RESTART AT 200 AFTER RECEIVING THE NEW DEVICE PARAMETERS.

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

4.3 PROGRAM AND/OR OPERATOR ACTION

1. LOAD THE PROGRAM INTO MEMORY.

2. SET STARTING ADDRESS ON THE SWITCH REGISTER

3. PRESS "LOAD ADDRESS".

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

5. PRESS "START".

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

EACH TEST FOUR TIMES ON EACH DRIVES PRESENT OR DRIVE
SELECTED BEFORE "END PASS" IS PRINTED. THE SECOND
AND SUBSEQUENT PASSES DO NOT NEED ANY OPERATOR INTERVENTION.

5.0 OPERATING PROCEDURE

5.1 OPERATIONAL SWITCH SETTINGS

IF THE PROGRAM IS BEING RUN ON A SWITCHLESS PROCESSOR (I. E.
AN 11/34) IT WILL DETERMINE THAT A HARDWARE SWTICH REGISTER
IS NOT PRESENT, AND WILL USE A "SOFTWARE" SWITCH REGISTER.
THE SETTINGS OF THE "SOFTWARE" SWITCHES ARE CONTROLLED
THROUGAH A KEYBOARD ROUTINE WHICH IS CALLED BY TYPING A
'CONTROL G'. THE PROGRAM WILL RECOGNIZE A 'CONTROL G' AT ANY
TIME EXCEPT WHEN IT IS AT A HIGHER PRIORITY PROCESSIHG A RP04/5/6
INTERRUPT. THE "SOFTWARE" SWITCH VALUEA ARE ENTERED AS AN
OCTAL NUMBER IN RESPONSE TO A PROMPT FROM THE SWITCH ENTRY
ROUTINE:

'SWR = NNNNNN NEW ='

EACH TIME SWITCH SETTINGS ARE ENTERED, THE ENTIRE SWITCH
REGISTER IMAGE MUST BE ENTERED. LEADING ZEROS ARE NOT
REQUIRED. 'RUBOUT' AND 'CONTROL U' FUNCTIONS MAY BE USED TO
CORRECT TYPING ERRORS DURING SWITCH ENTRY.

ON PROCESSORS WITH HARDWARE SWITCH REGISTERS, THE "SOFTWARE"
SWITCH REGISTER MAY ALSO BE USED. IF THE PROGRAM FINDS ALL
16 SWITCHES IN THE 'UP' POSITION WHEN IT IS STARTED, ALL
SWITCH REGISTER REFERENCES WILL BE TO THE "SOFTWARE" REGISTER
AND THE PROCEDURES DESCRIBED ABOVE MUST BE FOLLOWED.

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 12 - RH70 CONTROFLLER SELECT
                        THIS SWICH MUST BE SET AT THE START OF THE PROGRAM WHEN THE
                        DISK DRIVES TO BE TESTED ARE CONNECTED TO AN RH70
                        CONTROLLER.  IT MUST NOT BE SET WHEN DISK DRIVES TO BE TESTED
                        ARE CONNECTED TO AN RH11 CONTROLLER.

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

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

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


                        NOTE: ALSO SEE SECTION 8.3


                        SWITCH 8 - LOOP ON TEST IN SWR <7:0>
                        THIS IS A SPECIAL SWITCH.  WHEN SET SWITCHES 0 THRU 7
                        HAVE ONE MEANING AND WHEN RESET SWITCHES 0 THRU 7 HAVE
                        ANOTHER MEANING.  THIS MEANS THAT ANY SETTING OF SWITCH
                        0 THRU 7 MUST BE DONE WITH SWITCH 8 IN THE APPROPIATE
                        POSITION.  WHEN THIS SWITCH IS SET THEN SWITCHES 0 THRU
                        7 GIVE THE TEST NUMBER TO BE LOOPED ON.  FOR EXAMPLE
                        WITH SWITCH 8 SET AND SWITCH 3 SET THE PROGRAM WILL LOOP
                        ON TEST 10.  HOWEVER THIS SETTING MUST BE DONE AT THE
                        BEGINNING OF THE PROGRAM THEN ALL THE TESTS FROM 1 TO 10
                        WILL BE EXECUTED AND THEN TEST 10 WILL BE REPEATED OVER
                        AND OVER AGAIN.  WHEN THIS SWITCH IS NOT SET THEN SWITCHES
```

0 THRU 7 HAVE THE MEAING ITS NAME INDICATES.
FOR EXAMPLE SWITCH 7 IS "STOP FURTHER COMPARES: THAT IS
IF SWITCH 8 IS NOT SET AND SWITCH 7 IS SET THEN WHEN A
DATA ERROR IS DETECTED NO FURTHER COMPARES WILL BE DONE.
FOR EXAMPLE IN A 256 WORD BUFFER IF ALL THE WORDS ARE IN
ERROR THEN AFTER SEEING THE PRINTOUT FOR THE FIRST FEW
WORDS SETTING SWITCH 7 ONLY WILL STOP FURTHER PRINTOUTS
OF THIS ERROR AND GO ON WITH THE TEST RATHER THAN PRINT
ALL THE 256 WORDS. HOWEVER IF THIS WAS DONE WITH SWITCH 11
THEN THE NEXT ERROR THAT THE PROGRAM DETECTS IN A SUB-
SEQUENT TEST WILL ALSO BE LOST. BUT WITH SWITCH 7, ONLY
THIS GROUP OF DATA ERRORS ARE NOT PRINTED OUT. ANOTHER
EXAMPLE OF SWITCH 8 BEING LOW IS WITH SWITCH 6, WHICH
IS "ECC TEST-COMPARE END RESULT ONLY". THAT IS IF SWITCH
8 IS NOT SET AND SWITCH 6 IS SET THEN ON ECC TESTS (TEST
120 THRU TEST 134) INSTEAD OF COMPARING CONTENTS OF THE
POSITION REGISTER AND PATTERN REGISTER AFTER EVERY CLOCK,
COMPARES WILL ONLY BE DONE AT THE END OF ALL THE CLOCKS.

NOTE: ALSO SEE SECTION 8.3

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

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

5.2 SUB-ROUTINE ABSTRACTS

SEE SECTION 9 "SUBROUTINES"

6.0 ERRORS

ERROR PRINTOUTS CONTAIN THE ERROR ADDRESS AND OTHER
PERTINENT INFORMATION CONCERNING THE PARTICULAR FAILURE.
THIS INFORMATION MAY BE THE CONTENTS OF RELEVANT RP04/5/6

```
                        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.

                        SWITCH 12 MUST BE SET WHEN RUNNING ON AN RH70 CONTROLLER AND
                        IT MUST NOT BE SET WHEN RUNNING ON AN RH11 CONTROLLER.
                        BECAUSE OF THE REQUIREMENT FOR IT TO BE SET WHEN USING AN
                        RH70, THE PROGRAM CANNOT BE RUN IN CHAIN MODE WHEN USING THE
                        SOFTWARE SWITCH REGISTER FEATURE WHILE RUNNING ON AN RH70.
                        THIS IS BECAUSE THE ROUTINE WHICH GETS "SOFTWARE" SWTICH
                        SETTINGS IS NOT OPERABLE WHEN IN CHAIN MODE.

                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.

                        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.
                        @
```

```
                        ;*LAST REVISION 05-NOV-81
                        .TITLE  CZRJGE0 RP04/5/6 DSKLS PT1
                        ;*COPYRIGHT (C) 1976,1978,1981
                        ;*DIGITAL EQUIPMENT CORPORATION
                        ;*COLORADO SPGS., CO. 80919
                        ;*
                        ;*PROGRAM BY PETE BLACKSTONE
                        ;*
                        ;*THIS PROGRAM WAS ASSEMBLED USING THE PDP-11 MAINDEC SYSMAC
                        ;*PACKAGE (MAINDEC-11-DZQAC-C5), 18-MAR-81
                        ;*
                        .SBTTL  OPERATIONAL SWITCH SETTINGS
                        ;*
                        ;*      SWITCH                  USE
                        ;*      ------          --------------------
                        ;*        15            HALT ON ERROR
                        ;*        14            LOOP ON TEST
                        ;*        13            INHIBIT ERROR TYPEOUTS
                        ;*        12            RH70 CONTROLLER SELECT
                        ;*        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 *** 1100 ***
001100                  STACK   = 1100
104000                  ERROR   = EMT           ;;BASIC DEFINITION OF ERROR CALL
000004                  SCOPE   = IOT           ;;BASIC DEFINITION OF SCOPE CALL

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

                        ;*GENERAL PURPOSE REGISTER DEFINITIONS
000000                  R0      = %0            ;;GENERAL REGISTER
000001                  R1      = %1            ;;GENERAL REGISTER
000002                  R2      = %2            ;;GENERAL REGISTER
000003                  R3      = %3            ;;GENERAL REGISTER
000004                  R4      = %4            ;;GENERAL REGISTER
000005                  R5      = %5            ;;GENERAL REGISTER
000006                  R6      = %6            ;;GENERAL REGISTER
000007                  R7      = %7            ;;GENERAL REGISTER
000006                  SP      = %6            ;;STACK POINTER
000007                  PC      = %7            ;;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
001000                  SW9=SW09
000400                  SW8=SW08
000200                  SW7=SW07
000100                  SW6=SW06
000040                  SW5=SW05
000020                  SW4=SW04
000010                  SW3=SW03
000004                  SW2=SW02
000002                  SW1=SW01
000001                  SW0=SW00

                        ;*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
001000                  BIT9=BIT09
```

```
                000400          BIT8=BIT08
                000200          BIT7=BIT07
                000100          BIT6=BIT06
                000040          BIT5=BIT05
                000020          BIT4=BIT04
                000010          BIT3=BIT03
                000004          BIT2=BIT02
                000002          BIT1=BIT01
                000001          BIT0=BIT00

                                ;*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
    554
    555                         .SBTTL  RH11/RH70 REGISTERS
    556
    557                         ;*WORD COUNT REGISTER (RHWC)
    558                         ;*EACH BIT IS CALLED BY BIT NUMBER
    559
    560
    561                         ;*BUS ADDRESS REGISTER (RHBA)
    562                         ;*EACH BIT IS CALLED BY BIT NUMBER
    563
    564
    565                         ;*CONTROL AND STATUS REGISTER 2 (RHCS2)
    566
    567         000001          US1=    1               ;UNIT SELECT (BIT #0)
    568         000002          US2=    2               ;UNIT SELECT (BIT #1)
    569         000004          US4=    4               ;UNIT SELECT (BIT #2)
    570         000010          BAI=    10              ;BUS ADDRESS INCREMENT INHIBIT (BIT #3)
    571         000020          PAT=    20              ;INVERT PARITY ON MASS BUS TO EVEN (BIT #4)
    572         000040          CLR=    40              ;CLEAR (BIT #5)
    573         000100          IR=     100             ;INPUT READY (BIT #6)
    574         000200          OR=     200             ;OUTPUT READY (BIT #7)
    575         000400          MPE=    400             ;MASS BUS PARITY ERROR (BIT #8)
    576         001000          MXF=    1000            ;MISSED TRANSFER ERROR (BIT #9)
    577         002000          PGE=    2000            ;PROGRAM ERROR (BIT #10)
    578         004000          NEM=    4000            ;NON EXISTANT MEMORY (BIT #11)
    579         010000          NED=    10000           ;NON EXISTANT DRIVE (BIT #12)
    580         020000          UPE=    20000           ;UNIBUS PARITY ERROR (BIT #13)
    581         040000          WCE=    40000           ;WRITE CHECK ERROR (BIT #14)
    582         100000          DLT=    100000          ;DATA LATE (BIT #15)
    583
    584
    585                         ;*DATA BUFFER REGISTER (RHDB)
    586                         ;*EACH BIT IS CALLED BY BIT NUMBER
    587
```

```
                                ;*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)
002000                          PSEL=   2000            ;PORT SELECT (BIT #10)
004000                          DVA=    4000            ;DEVICE AVAILABLE (BIT #11)
020000                          MCPE=   20000           ;MASSBUSS PARITY ERROR (BIT #13)
040000                          TRE=    40000           ;TRANSFER ERROR (BIT #14)
100000                          SC=     100000          ;SPECIAL CONDITION (BIT #15)
                                ;*STATUS REGISTER (RHDS1) (#01)
000001                          DF5=    1               ;DRIVE FORWARD 5''/SEC.  (BIT #0)
000002                          DFF20=  2               ;DRIVE FORWARD 20''/SEC.  (BIT #1)
000004                          DIGB=   4               ;DRIVE TO INNER GAVRD BAND (BIT #2)
000010                          GRV=    10              ;GO REVERSE (BIT #3)
000020                          DL64=   20              ;DIFFERENCE LESS THAN 64 (BIT #4)
000040                          DE1=    40              ;DIFFERENCE EQUALS 1 (BIT #5)
000100                          VV=     100             ;VOLUME VALID (BIT #6)
000200                          DRY=    200             ;DRIVE READY (BIT #7)
000400                          DPR=    400             ;DRIVE PRESENT (BIT #8)
001000                          PROG=   1000            ;PROGRAMABLE (BIT #9)
002000                          LST=    2000            ;LAST SECTOR TRANSFERRED (BIT #10)
004000                          WRL=    4000            ;WRITE LOCK (BIT #11)
010000                          MOL=    10000           ;MEDIUM ON-LINE (BIT #12)
020000                          PIP=    20000           ;POSITIONING OPERATION IN PROGRESS (BIT #13)
040000                          ERR=    40000           ;COMPOSIT ERROR. (BIT #14)
100000                          ATA=    100000          ;ATTENTION ACTIVE (BIT #15)
                                ;*ERROR REGISTER #01 (RHER1) (#02)
000001                          ILF=    1               ;ILLEGAL FUNCTION (BIT #0)
000002                          ILR=    2               ;ILLEGAL REGISTER (BIT #1)
000004                          RMR=    4               ;REGISTER MODIFICATION REFUSED (BIT #2)
000010                          PAR=    10              ;PARITY ERROR (BIT #3)
000020                          FER=    20              ;FORMAT ERROR (BIT #4)
000040                          WCF=    40              ;WRITE CLOCK FAIL (BIT #5)
000100                          ECH=    100             ;ECC HARD ERROR (BIT #6)
000200                          HCE=    200             ;HEADER COMPARE ERROR (BIT #7)
000400                          HCRC=   400             ;HEADER CRC ERROR (BIT #8)
001000                          AOE=    1000            ;ADDRESS OVERFLOW ERROR (BIT #9)
002000                          IAE=    2000            ;INVALID ADDRESS ERROR (BIT #10)
004000                          WLE=    4000            ;WRITE LOCK ERROR (BIT #11)
010000                          DTE=    10000           ;DRIVE TIMING ERROR (BIT #12)
020000                          OPI=    20000           ;OPERATION INCOMPLETE (BIT #13)
040000                          UNS=    40000           ;DRIVE UNSAFE (BIT #14)
100000                          DCK=    100000          ;DATA CHECK ERROR (BIT 15)
                                ;*MAINTAINABILITY REGISTER (RHMR)(#03)
000001                          DMD=    1               ;DIAGINOSTIC MODE (BIT #0)
```

```
                                MCLK=   2                       ;MAINTAINABILITY CLOCK (BIT #1)
                                MINX=   4                       ;MAINTAINABILITY INDEX (BIT #2)
                                MSTCK=  10                      ;MAINTAINABILITY SECTOR CLOCK (BIT #3)
                                MRD=    20                      ;MAINTAINABILITY READ (BIT #4)
                                MWR=    40                      ;MAINTAINABILITY WRITE (BIT #5)
                                DENVL=  200                     ;DATA ENVELOPE (BIT #7)
                                ZER=    400                     ;ZERO DETECT (BIT #8)
                                DTSY=   1000                    ;MAINTAINABILITY SYNC DETECTED (BIT #9)

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

                                AT0=    1                       ;DEVICE 0 (BIT #0)
                                AT1=    2                       ;DEVICE 1 (BIT #1)
                                AT2=    4                       ;DEVICE 2 (BIT #2)
                                AT3=    10                      ;DEVICE 3 (BIT #3)
                                AT4=    20                      ;DEVICE 4 (BIT #4)
                                AT5=    40                      ;DEVICE 5 (BIT #5)
                                AT6=    100                     ;DEVICE 6 (BIT #6)
                                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)

                                EXT1=   1                       ;EXTENSION 1 (BIT #0)
                                EXT2=   2                       ;EXTENSION 2 (BIT #1)
                                EXT4=   4                       ;EXTENSION 3 (BIT #2)
                                EXT10=  10                      ;EXTENSION 4 (BIT #3)
                                EXT20=  20                      ;EXTENSION 5 (BIT #4)
                                EXT40=  40                      ;EXTENSION 6 (BIT #5)
                                SC1=    100                     ;SECTOR COUNT FIELD 0 (BIT #6)
                                SC2=    200                     ;SECTOR COUNT FIELD 1 (BIT #7)
                                SC4=    400                     ;SECTOR COUNT FIELD 2 (BIT #8)
                                SC10=   1000                    ;SECTOR COUNT FIELD 3 (BIT #9)
                                SC20=   2000                    ;SECTOR COUNT FIELD 4 (BIT #10)
                                TRK1=   4000                    ;TRACK FIELD 1 (BIT #11)
                                TRK2=   10000                   ;TRACK FIELD 2 (BIT #12)
                                TRK4=   20000                   ;TRACK FIELD 3 (BIT #13)
                                TRK10=  40000                   ;TRACK FIELD 4 (BIT #14)
                                TRK20=  100000                  ;TRACK FIELD 5 (BIT #15)

                                ;*RP04 ERROR REGISTER #2 (RHER2) (#10)

                                WCU=    1                       ;WRITE CURRENT UNSAFE (BIT #0)
                                CSF=    2                       ;CURRENT SINK FAILURE (BIT #1)
                                WSU=    4                       ;WRITE SELECT UNSAFE (BIT #2)
                                CSU=    10                      ;CURRENT SWITCH UNSAFE (BIT #3)
                                MSE=    20                      ;MOTOR SEQUENCE ERROR (BIT #4)
                                TDF=    40                      ;TRANSITIONS DETECTOR FAILURE (BIT #5)
                                TUF=    100                     ;TRANSITIONS UNSAFE (BIT #6)
```

```
                                ;FAILSAFE ENABLED (BIT #7)
000200          FEN=    200     ;WRITE READY UNSAFE (BIT #8)
000400          WRU=    400     ;MULTIPLE HEAD SELECT (BIT #9)
001000          MHS=    1000    ;NO HEAD SELECTION (BIT #10)
002000          NHS=    2000    ;INDEX ERROR (BIT #11)
004000          IXE=    4000    ;30VOLT UNSAFE (BIT #12)
010000          VU30=   10000   ;PLO UNSAFE (BIT #13)
020000          PLU=    20000   ;ACUNSAFE (BIT #15)
100000          ACU=    100000

                ;*RP05/6 ERROR REGISTER #2 (RHER2) (#10)

000001          WCU=    1       ;WRITE CURENT UNSAFE
000002          CSF=    2       ;CURRENT SINK FAILURE
000004          WSU=    4       ;CURENT SELECT UNSAFE
000010          CSU=    10      ;CURRENT SWITCH UNSAFE
000020          RAW=    20      ;READ AND WRITE
000040          TDF=    40      ;TRANSITIONS DETECTOR FAILURE
000100          TUF=    100     ;TRANSITIONS UNSAFE
000200          ABS=    200     ;ABNORMAL STOP
000400          WRU=    400     ;WRITE READY UNSAFE
001000          MHS=    1000    ;MULTIPLE HEAD SELECT
002000          NHS=    2000    ;NO HEAD SELECTION
004000          IXE=    4000    ;INDEX ERROR
020000          PLU=    20000   ;PLO UNSAFE

                ;*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)

000001          PSU=    1       ;PACK SPEED UNSAFE (BIT #0)
000002          VUF=    2       ;VELOCITY UNSAFE (BIT #1)
000010          UWR=    10      ;ANY UNSAFE EXCEPT READ/WRITE (BIT #3)
```

RH11/RH70 REGISTERS

```
000020                  PRE=    20                      ;DISK PACK ROTATION ERROR (BIT #4)
000040                  ACL=    40                      ;AC LOW (BIT #5)
000100                  OPDCL=  100                     ;DC LOW (BIT #6)
040000                  SKI=    40000                   ;SEEK INCOMPLETE (BIT #14)
100000                  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  MEMORY MANAGEMENT DEFINITIONS

                        ;*KT11 VECTOR ADDRESS

000250                  MMVEC   = 250

                        ;*KT11 STATUS REGISTER ADDRESSES

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

                        ;*KERNEL "I" PAGE DESCRIPTOR REGISTERS

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

                        ;*KERNEL "I" PAGE ADDRESS REGISTERS

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

774
```

```
      1                                 .SBTTL  TRAP CATCHER

                000000                          .=0
                                        ;*ALL UNUSED LOCATIONS FROM 4 - 776 CONTAIN A ".+2,HALT"
                                        ;*SEQUENCE TO CATCH ILLEGAL TRAPS AND INTERRUPTS
                                        ;*LOCATION 0 CONTAINS 0 TO CATCH IMPROPERLY LOADED VECTORS
                000174                  .=174
      000174    000000                DISPREG: .WORD  0               ;;SOFTWARE DISPLAY REGISTER
      000176    000000                SWREG:   .WORD  0               ;;SOFTWARE SWITCH REGISTER

                                        .SBTTL  STARTING ADDRESS(ES)

      000200    000137  004310                JMP     @#BEGIN         ;;JUMP TO STARTING ADDRESS OF PROGRAM
    2 000204    000137  045104                JMP     @#BASECH        ;MODIFY ADDRESSES
    3 000210    000137  004316                JMP     @#BEGIN2        ;SELECT DRIVE START
    4
    5                                   .SBTTL  ACT11 HOOKS

                                        ;;*****************************************************************
                                        ;HOOKS REQUIRED BY ACT11
                000214                          $SVPC=.                 ;SAVE PC
                000046                          .=46
      000046    040252                          $ENDAD                  ;;1)SET LOC.46 TO ADDRESS OF $ENDAD IN .$EOP
                000052                          .=52
      000052    020000                          .WORD   20000           ;;2)SET LOC.52 TO 20000
                000214                          .=$SVPC                 ;; RESTORE PC
      6
```

```
      0                         .SBTTL  COMMON TAGS

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

            001100                      .=1100
      001100                    $CMTAG:                         ;;START OF COMMON TAGS
      001100  000000            $PASS:  .WORD   0               ;;CONTAINS PASS COUNT
      001102     000            $TSTNM: .BYTE   0               ;;CONTAINS THE TEST NUMBER
      001103     000            $ERFLG: .BYTE   0               ;;CONTAINS ERROR FLAG
      001104  000000            $ICNT:  .WORD   0               ;;CONTAINS SUBTEST ITERATION COUNT
      001106  000000            $LPADR: .WORD   0               ;;CONTAINS SCOPE LOOP ADDRESS
      001110  000000            $LPERR: .WORD   0               ;;CONTAINS SCOPE RETURN FOR ERRORS
      001112  000000            $ERTTL: .WORD   0               ;;CONTAINS TOTAL ERRORS DETECTED
      001114     000            $ITEMB: .BYTE   0               ;;CONTAINS ITEM CONTROL BYTE
      001115     001            $ERMAX: .BYTE   1               ;;CONTAINS MAX. ERRORS PER TEST
      001116  000000            $ERRPC: .WORD   0               ;;CONTAINS PC OF LAST ERROR INSTRUCTION
      001120  000000            $GDADR: .WORD   0               ;;CONTAINS ADDRESS OF 'GOOD' DATA
      001122  000000            $BDADR: .WORD   0               ;;CONTAINS ADDRESS OF 'BAD' DATA
      001124  000000            $GDDAT: .WORD   0               ;;CONTAINS 'GOOD' DATA
      001126  000000            $BDDAT: .WORD   0               ;;CONTAINS 'BAD' DATA
      001130  000000                    .WORD   0               ;;RESERVED--NOT TO BE USED
      001132  000000                    .WORD   0
      001134     000            $AUTOB: .BYTE   0               ;;AUTOMATIC MODE INDICATOR
      001135     000            $INTAG: .BYTE   0               ;;INTERRUPT MODE INDICATOR
      001136  000000                    .WORD   0
      001140  177570            SWR:    .WORD   DSWR            ;;ADDRESS OF SWITCH REGISTER
      001142  177570            DISPLAY: .WORD  DDISP           ;;ADDRESS OF DISPLAY REGISTER
      001144  177560            $TKS:   177560                  ;;TTY KBD STATUS
      001146  177562            $TKB:   177562                  ;;TTY KBD BUFFER
      001150  177564            $TPS:   177564                  ;;TTY PRINTER STATUS REG. ADDRESS
      001152  177566            $TPB:   177566                  ;;TTY PRINTER BUFFER REG. ADDRESS
      001154     000            $NULL:  .BYTE   0               ;;CONTAINS NULL CHARACTER FOR FILLS
      001155     002            $FILLS: .BYTE   2               ;;CONTAINS # OF FILLER CHARACTERS REQUIRED
      001156     012            $FILLC: .BYTE   12              ;;INSERT FILL CHARS. AFTER A 'LINE FEED''
      001157     000            $TPFLG: .BYTE   0               ;;'TERMINAL AVAILABLE'' FLAG (BIT<07>=0=YES)
      001160  000000            $REGAD: .WORD   0               ;;CONTAINS THE ADDRESS FROM
                                                                ;;WHICH  ($REG0) WAS OBTAINED
      001162  000000            $REG0:  .WORD   0               ;;CONTAINS (($REGAD)+0)
      001164  000000            $REG1:  .WORD   0               ;;CONTAINS (($REGAD)+2)
      001166  000000            $REG2:  .WORD   0               ;;CONTAINS (($REGAD)+4)
      001170  000000            $REG3:  .WORD   0               ;;CONTAINS (($REGAD)+6)
      001172  000000            $REG4:  .WORD   0               ;;CONTAINS (($REGAD)+10)
      001174  000000            $REG5:  .WORD   0               ;;CONTAINS (($REGAD)+12)
      001176  000000            $TMP0:  .WORD   0               ;;USER DEFINED
      001200  000000            $TMP1:  .WORD   0               ;;USER DEFINED
      001202  000000            $TMP2:  .WORD   0               ;;USER DEFINED
      001204  000000            $TMP3:  .WORD   0               ;;USER DEFINED
      001206  000000            $TMP4:  .WORD   0               ;;USER DEFINED
      001210  000000            $TMP5:  .WORD   0               ;;USER DEFINED
      001212  000000            $TIMES: 0                       ;;MAX. NUMBER OF ITERATIONS
      001214  000000            $ESCAPE:0                       ;;ESCAPE ON ERROR ADDRESS
      001216     207     377 377 $BELL:  .ASCIZ  <207><377><377> ;;CODE FOR BELL
      001222     077            $QUES:  .ASCII  /?/             ;;QUESTION MARK
      001223     015            $CRLF:  .ASCII  <15>            ;;CARRIAGE RETURN
      001224     012     000    $LF:    .ASCIZ  <12>            ;;LINE FEED
```

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

```
                                .SBTTL  USER DEFINED TAGS

001226  000254                  RPVEC:  254                     ;RP04 VECTOR ADDRESS
001230  176722                  RHDB:   176722                  ;DATA BUFFER SEE NOTE ABOVE
001232  176702                  RHWC:   176702                  ;WORD COUNT SEE NOTE ABOVE
001234  176704                  RHBA:   176704                  ;BUS ADDRESS SEE NOTE ABOVE
001236  176710                  RHCS2:  176710                  ;CONTROL AND STATUS 2 SEE NOTE ABOVE
001240  176700                  RHCS1:  176700                  ;CONTROL AND STATUS 1 SEE NOTE ABOVE
001242  176714                  RHER1:  176714                  ;ERROR #1 SEE NOTE ABOVE
001244  176706                  RHDST:  176706                  ;DESIRED SECTOR/TRACK ADDRESS SEE NOTE ABOVE
001246  176740                  RHER2:  176740                  ;ERROR #2 SEE NUTE ABOVE
001250  176732                  RHOF:   176732                  ;OFFSET SEE NOTE ABOVE
001252  176734                  RHCA:   176734                  ;DESIRED CYLINDER ADDRESS SEE NOTE ABOVE
001254  176742                  RHER3:  176742                  ;ERROR #3 SEE NOTE ABOVE
001256  176716                  RHAS:   176716                  ;ATTENTION SUMMARY SEE NOTE ABOVE
001260  176724                  RHMR:   176724                  ;MAINTAINABILITY SEE NOTE ABOVE
001262  176712                  RHDS1:  176712                  ;DRIVE STATUS SEE NOTE ABOVE
001264  176726                  RHDT:   176726                  ;DRIVE TYPE SEE NOTE ABOVE
001266  176730                  RHSN:   176730                  ;SERIAL NUMBER SEE NOTE ABOVE
001270  176744                  RHEC1:  176744                  ;ECC POSITION SEE NOTE ABOVE
001272  176746                  RHEC2:  176746                  ;ECC PATTERN SEE NOTE ABOVE
001274  176720                  RHLA:   176720                  ;LOOK-AHEAD SEE NOTE ABOVE
001276  176736                  RHCC:   176736                  ;CURRENT CYLINDER ADDRESS SEE NOTE ABOVE

                                ;ADDITIONAL REGISTERS LOCATED IN THE RH70 CONTROLLER LOGIC

001300  176750                  RHBAE:  176750                  ;BUS ADDRESS EXTENSION REGISTER
001302  176752                  RHCS3:  176752                  ;CONTROL AND STATUS REGISTER #3

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

001304  000000                  DB:     0                       ;DATA BUFFER
001306  000000                  WC:     0                       ;WORD COUNT
001310  000000                  BA:     0                       ;BUS ADDRESS
001312  000000                  CS2:    0                       ;CONTROL AND STATUS 2

001314  000000                  CS1:    0                       ;CONTROL AND STATUS 1
001316  000000                  ER1:    0                       ;ERROR #1
001320  000000                  DST:    0                       ;DESIRED SECTOR/TRACK ADDRESS
001322  000000                  ER2:    0                       ;ERROR #2
001324  000000                  OF:     0                       ;OFFSET
001326  000000                  CA:     0                       ;DESIRED CYLINDER ADDRESS
001330  000000                  ER3:    0                       ;ERROR #3
001332  000000                  AS:     0                       ;ATTENTION SUMMARY
001334  000000                  MR:     0                       ;MAINTAINABILITY
001336  000000                  DS1:    0                       ;DRIVE STATUS
001340  000000                  DT:     0                       ;DRIVE TYPE
001342  000000                  SN:     0                       ;SERIAL NUMBER
001344  000000                  EC1:    0                       ;ECC POSITION
001346  000000                  EC2:    0                       ;ECC PATTERN
001350  000000                  LA:     0                       ;LOOK-AHEAD
001352  000000                  CC:     0                       ;CURRENT CYLINDER ADDRESS

                                ;FLAGS & INTERNAL PROGRAM CONTROL WORDS
```

USER DEFINED TAGS

```
001354                  UNITS:  .BLKW   8.      ;TABLE OF DRIVES PRESENT TO TEST
001374  000000          UNIT:   .WORD   0       ;UNIT UNDER TEST
001376  000000          NOUNIT: .WORD   0       ;NUMBER OF UNITS PRESENT
                                                ;USED TO KEEP TRACK OF UNIT UNDER TEST
001400  000000          NUNIT:  .WORD   0       ;USED TO DETERMINE IF THERE IS MORE
                                                ;THAN ONE UNIT
001402  000000          SELECT: .WORD   0       ;ALL ONES INDICATE UNIT TO BE SELECTED
001404  000000          UNITSL: .WORD   0       ;UNIT NO. SELECTED
001406  000000          ERFLG$: 0               ;ERROR FLAG
001410  000000          SAVDT:  0               ;SAVE DRIVE TYPE REGISTER
                                                ;FOR COMPARISON IN DRIVE CLEAR TEST
                                                ;AND RH INIT TEST
001412  000000          SAVSN:  0               ;SAVE SERIAL NUMBER REGISTER
                                                ;FOR COMPARISON IN DRIVE CLEAR TEST
                                                ;AND RH INIT TEST
001414  000000          PCJSR:  0               ;SAVE PC OF JSR WHICH GAVE THE ERROR
001416  000000          ATTENT: 0               ;ATTENTION BIT FOR PRESENT UNIT
001420  000000          TOTALAT: 0              ;TATAL ATTENTION BITS
001422  000000          TMPILL: 0               ;TEMPORARY ILLEGAL FUNCTION
001424  000000          TSECC:  0               ;FLAG TO SAY IF ECC TEST OR NOT
                                                ;WHEN =177777 IT IS AN ECC TEST
                                                ;WHEN =0IT IS NOT AN ECC TEST
001426  000000          TESDTE: 0               ;FLAG TO SAY IF DRIVE TIMING ERROR OR NOT
                                                ;WHEN = 177777 IT IS A DTE TEST
                                                ;WHEN = 0      IT IS NOT A DTE TEST
001430  000000          TAGDTE: 0               ;TEMPORARY TAG USED IN DRIVE TIMING
                                                ;ERROR TEST
001432  000000          TSTNM:  0               ;TEST NUMBER
001434  000000          RP06:   0               ;IF 0 PROGRAM WILL TREAT DRIVE AS RP04
001436  000000          RH70:   0               ;IF 1 PROGRAM IS RUNNING ON RH70
                                                ;IF 0 PROGRAM IS ON AN RH11
001440  000000          SILOSZ: .WORD   0       ;RH SILO SIZE

                        ;FUNCTION EQUATES

                        ;TABLE OF FUNCTIONS FOR RHCS1, THEN "GO" BIT HAS TO BE SET

001442                  FUTABL:
001442  000000          NOPERA: 0               ;NO OPERATION
001444  000002          UNLOAD: 2               ;UNLOAD (STAND BY)
001446  000006          RECALI: 6               ;RECALIBRATE
001450  000010          DCLEAR: 10              ;DRIVE CLEAR
001452  000012          RELEAS: 12              ;RELEASE (DUAL-PORT OPERATION)
001454  000030          SERCH:  30              ;SEARCH COMMAND
001456  000050          WRCHEK: 50              ;WRITE CHECK DATA
001460  000052          WRCHDT: 52              ;WRITE CHECK HEADER AND DATA
001462  000060          WRIDAT: 60              ;WRITE DATA
001464  000062          WRIFOR: 62              ;WRITE HEADER AND DATA (FORMAT)
001466  000070          READAT: 70              ;READ DATA
001470  000072          REFOR:  72              ;READ HEADER AND DATA
001472  000004          SEECOM: 4               ;SEEK COMMAND
001474  000014          OFSETC: 14              ;OFFSET COMMAND
001476  000016          RETCL:  16              ;RETURNTO CENTERLINE
001500  000022          PKACK:  22              ;PACK ACKNOWLEDGE
001502  000020          READIN: 20              ;READ IN
001504  000000          ILLEGL: .WORD           ;COMPUTED ILLEGAL FUNCTION
```

USER DEFINED TAGS

```
                                ;DATA BUFFERS FOR READ WRITE

001510                          WRFROM: .BLKW   274.            ;WRITE FROM THIS BUFFER
002554                          REINTO: .BLKW   274.            ;READ INTO THIS BUFFER

                                ;TABLE FOR ATTENTION BITS ATTENTION TABLE

003620  001  002  004  ATABLE: .BYTE   1,2,4,10,20,40,100,200
```

```
                                .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

   003630                       $ERRTB:
 1
 2                              ;ERROR 1
 3
 4 003630  057072                       EM1             ;WRONG DATA IN READING OR WRITING HARDWARE REGISTER
 5 003632  063205                       DH1             ;PC
 6                                                      ;REG. ADDR.
 7                                                      ;GOOD DATA
 8                                                      ;RECEIVED DATA
 9 003634  067540                       DT1             ;$ERRPC,REGADR,$GDDAT,$BDDAT
10 003636  070256                       DF1             ;0,0,0,0,0
11
12                              ;ERROR 2
13
14 003640  057155                       EM2             ;ERROR ON  DATA COMMAND
15
16 003642  066323                       DH33            ;PC
17                                                      ;PC OF JSR
18                                                      ;TEST NO
19                                                      ;WORD NO.
20                                                      ;GOOD DATA
21                                                      ;CONTENTS OF RHCS1
22                                                      ;CONTENTS OF RHDS1
23                                                      ;CONTENTS OF RHER1
24 003644  070120                       DT33            ;$ERRPC,PCJSR,$TSTNM,ERWORD,$GDDAT,CS1,DS1,ER1
25 003646  070426                       DF33            ;0,0,0,1,0,0,0,0
26
27                              ;ERROR 3
28
29 003650  057155                       EM2             ;ERROR ON  DATA COMMAND
30
31 003652  066101                       DH32            ;PC
32                                                      ;PC OF JSR
33                                                      ;TEST NO
34                                                      ;WORD NO.
35                                                      ;GOOD DATA
36                                                      ;BAD DATA
37                                                      ;CONTENTS OF RHCS1
38                                                      ;CONTENTS OF RHDS1
39                                                      ;CONTENTS OF RHER1
40
41 003654  070074                       DT32            ;$ERRPC,PCJSR,$TSTNM,ERWORD,$GDDAT,$BDDAT,CS1,DS1,ER1
42 003656  070415                       DF32            ;0,0,0,1,0,0,0,0,0,
```

ERROR POINTER TABLE

```
                                ;ERROR 4
003660  057155                          EM2             ;ERROR ON  DATA COMMAND
003662  065677                          DH31            ;PC
                                                        ;TEST NO
                                                        ;WORD NO.
                                                        ;GOOD DATA
                                                        ;BAD DATA
                                                        ;CONTENTS OF RHCS1
                                                        ;CONTENTS OF RHDS1
                                                        ;CONTENTS OF RHER1
003664  070052                          DT31            ;$ERRPC,$TSTNM,ERWORD,$GDDAT,$BDDAT,CS1,DS1,ER1
003666  070405                          DF31            ;0,0,1,0,0,0,0,0,
                                ;ERROR 5
003670  000000                          0               ;
003672  000000                          0               ;
003674  070052                          DT31            ;$ERRPC,$TSTNM,ERWORD,$GDDAT,$BDDAT,CS1,DS1,ER1
003676  070405                          DF31            ;0,0,1,0,0,0,0,0,
                                ;ERROR 6
003700  057204                          EM6             ;ERROR ON WRITE HEADER AND DATA
003702  066101                          DH32            ;PC
                                                        ;PC OF JSR
                                                        ;TEST NO
                                                        ;WORD NO.
                                                        ;GOOD DATA
                                                        ;BAD DATA
                                                        ;CONTENTS OF RHCS1
                                                        ;CONTENTS OF RHDS1
                                                        ;CONTENTS OF RHER1
003704  070074                          DT32            ;$ERRPC,PCJSR,$TSTNM,ERWORD,$GDDAT,$BDDAT,CS1,DS1,ER1
003706  070415                          DF32            ;0,0,0,1,0,0,0,0,0,
                                ;ERROR 7
003710  057204                          EM6             ;ERROR ON WRITE HEADER AND DATA
003712  063327                          DH2             ;PC
                                                        ;TEST NO
                                                        ;WORD NO.
                                                        ;GOOD DATA
                                                        ;BAD DATA
003714  067566                          DT3             ;$ERRPC,$TSTNM,ERWORD,$GDDAT,$BDDAT
003716  070267                          DF3             ;0,0,1,0,0,
                                ;ERROR 10
003720  000000                          0               ;
003722  000000                          0               ;
003724  067566                          DT3             ;$ERRPC,$TSTNM,ERWORD,$GDDAT,$BDDAT
```

ERROR POINTER TABLE

```
003726  070267                          DF3             ;0,0,1,0,0,

                                ;ERROR 11

003730  057243                          EM11            ;CONTROLLER OR DRIVE STATUS
003732  063451                          DH11            ;PC
                                                        ;TEST NO
                                                        ;FAILING REG. ADDR
                                                        ;CONTENTS OF RHCS1
                                                        ;CONTENTS OF RHCS2
                                                        ;CONTENTS OF RHDS1
                                                        ;CONTENTS OF RHER1
003734  067602                          DT11            ;$ERRPC,$TSTNM,$BDADR,CS1,CS2,DS1,ER1
003736  070274                          DF11            ;0,0,0,0,0,0

                                ;ERROR 12

003740  057243                          EM11            ;WRONG DATA FROM SILO

003742  063205                          DH1             ;PC
                                                        ;REG.ADDR
                                                        ;GOOD DATA
                                                        ;RECEIVED DATA
003744  067540                          DT1             ;$ERRPC,REGADR,$GDDAT,$BDDAT
003746  070256                          DF1             ;0,0,0,0

                                ;ERROR 13

003750  000000                          0
003752  000000                          0
003754  067540                          DT1             ;$ERRPC,TSTNM,REGADR,$GDDAT,$BDDAT
003756  070256                          DF1             ;0,0,0,0,0

                                ;ERROR 14

003760  057276                          EM14            ;REGISTER FAILED
003762  063630                          DH14            ;PC
                                                        ;FAILING REG. ADDR
                                                        ;CONTENTS OF FAILING REG.
                                                        ;CONTENTS OF RHCS1
                                                        ;CONTENTS OF RHCS2
                                                        ;CONTENTS OF RHDS1
                                                        ;CONTENTS OF RHER1
003764  067622                          DT14            ;$ERRPC,$BDADR,$BDDAT,CS1,CS2,DS1,ER1
003766  070303                          DF14            ;0,0,0,0,0,0,0

                                ;ERROR 15

003770  057316                          EM15            ;SPECIFIED REG. NON EXISTANT SO ABORT
                                                        ;PROGRAM
003772  064027                          DH15            ;PC
                                                        ;ADDR. OF REG
003774  067644                          DT15            ;$ERRPC,TEMP1
003776  070313                          DF15            ;0,0

                                ;ERROR 16
```

ERROR POINTER TABLE

```
157 004000  057366                        EM16            ;WAIT LOOP FAILED
158 004002  064060                        DH16            ;PC
159                                                       ;WAT PC
160                                                       ;BIT WANTED
161                                                       ;REG. ADR.
162                                                       ;REG. CONT.
163 004004  067654                        DT16            ;$ERRPC,$TMP3,$TMP1,$TMP0,$BDDAT
164 004006  070316                        DF16            ;0,0,0,0
165
166                             ;ERROR 17
167
168 004010  057407                        EM17            ;WRITE CHECK FAILING
169 004012  064222                        DH17            ;PC
170                                                       ;TEST NO
171                                                       ;CONTENTS OF RHBA
172                                                       ;CONTENTS OF RHDB
173                                                       ;CONTENTS OF RHWC
174                                                       ;CONTENTS OF RHCS1
175                                                       ;CONTENTS OF RHCS2
176 004014  067672                        DT17            ;$ERRPC,$TSTNM,$BA,DB,WC,CS1,CS2
177 004016  070323                        DF17            ;0,0,0,0,0,0,0
178
179                             ;ERROR 20
180
181 004020  057433                        EM20            ;REGISTER FAILING
182 004022  064404                        DH20            ;PC
183                                                       ;TST NO
184                                                       ;CONTENTS OF RHER1
185                                                       ;CONTENTS OF RHER2
186                                                       ;CONTENTS OF RHER3
187                                                       ;CONTENTS OF RHAS
188                                                       ;CONTENTS OF RHDS1
189 004024  067712                        DT20            ;$ERRPC,TSTNM ER1,ER2,ER3,AS,DS1
190 004026  070332                        DF20            ;0,0,0,0,0,0,0
191
192                             ;ERROR 21
193
194 004030  057454                        EM21            ;INTERRUPT FAILING
195 004032  064566                        DH21            ;PC
196                                                       ;TEST NO
197                                                       ;CONTENTS OF RHCS1
198                                                       ;CONTENTS OF RHAS
199                                                       ;CONTENTS OF RHDS1
200 004034  067732                        DT21            ;$ERRPC,TSTNM,CS1,AS,DS1
201 004036  070341                        DF21            ;0,0,0,0,0
202
203                             ;ERROR 22
204
205 004040  057476                        EM22            ;ERROR IN DRIVE PRESENT -
206                                                       ;LOOKING AT RHAS AND RHCS2-NED(BIT#12)
207                                                       ;DRIVES PRESENT DO NOT AGREE
208                                                       ;NOTE: ON DUAL PORT SYSTEM
209                                                       ;DRIVE ON OTHER PORT WILL NOT GIVE NED
210                                                       ;HENCE THERE WILL BE A MISSMATCH
211 004042  064705                        DH22            ;PC
212                                                       ;TEST NO
213                                                       ;RHAS UNIT (RHER1 BITS SET)
```

ERROR POINTER TABLE

```
                                                        ;RHCS2 UNIT ('NED' BIT TEST)
004044  067746                          DT22            ;$ERRPC,TSTNM
004046  070346                          DF22            ;0,0

                                ;ERROR 23

004050  000000                          0               ;NO LONGER USED DUE TO SPECIAL 'NED'
004052  000000                          0               ;TEST TABLE TYPE OUT ROUTINE
004054  000000                          0
004056  000000                          0

                                ;ERROR  24

004060  060147                          EM24            ;LOOK AHEAD REGISTER AT THE
                                                        ;BEGINNING OF A SECTOR IS IN
                                                        ;ERROR
004062  065007                          DH24            ;PC
                                                        ;RHDST
                                                        ;BAD RHLA
                                                        ;GOOD RHLA
                                                        ;SECTOR NO
                                                        ;SECTOR CLOCK
004064  067754                          DT24            ;$ERRPC,DST,$BDDAT,$TMP1,$TMP2,$TMP3
004066  070352                          DF24            ;0,0,0,0,0

                                ;ERROR  25

004070  060242                          EM25            ;LOOK AHEAD REGISTER IS
                                                        ;IN ERROR
004072  065007                          DH24            ;PC
                                                        ;RHDST
                                                        ;BAD RHLA
                                                        ;GOOD RHLA
                                                        ;SECTOR NO
                                                        ;SECTOR CLOCK
004074  067754                          DT24            ;$ERRPC,DST,$BDDAT,$TMP1,$TMP2,$TMP3
004076  070352                          DF24            ;0,0,0,0,0

                                ;ERROR 26

004100  057243                          EM11            ;CONTROLLER OR DRIVE STATUS

004102  065171                          DH26            ;PC
                                                        ;PC OF JSR
                                                        ;FAILING REGISTER ADDRESS
                                                        ;CONTENTS OF RHCS1
                                                        ;CONTENTS OF RHCS2
                                                        ;CONTENTS OF RHDS1
                                                        ;CONTENTS OF RHER1

004104  067774                          DT26            ;$ERRPC,PCJSR,$BDADR,CS1,CS2,DS1,ER1
004106  070361                          DF26            ;0,0,0,0,0,0,

                                ;ERROR 27

004110  057072                          EM1             ;ERROR IN READING OR WRITING HARDWARE REGISTER
```

ERROR POINTER TABLE

```
                                                                ;PC
004112  065373                        DH27                    ;PC OF JSR
                                                                ;TEST NUMBER
                                                                ;FAILING REGISTER
                                                                ;GOOD DATA
                                                                ;RECEIVED DATA

004114  070016                        DT27                    ;$ERRPC,PCJSR,TSTNM,REGADR,$GDDAT,$BDDAT
004116  070371                        DF27                    ;0,0,0,0,0,0

                              ;ERROR 30

004120  060302                        EM30                    ;CURRENT CYLINDER DOES NOT REFLECT DESIRED CYLINDER REG.
004122  065535                        DH30                    ;PC
                                                                ;PC OF JSR
                                                                ;REGISTER ADDRESS
                                                                ;GOOD DATA
                                                                ;BAD DATA

004124  070034                        DT30                    ;$ERRPC,PCJSR,REGADR,$GDDAT,$BDDAT
004126  070377                        DF30                    ;0,0,0,0,0

                              ;ERROR 31

004130  060422                        EM31                    ;ECC GENERATED IS INCORRECT
                                                                ;EVERY WORD IN THIS SECTOR IS GIVEN IN 'DATA USED'

004132  066525                        DH34                    ;PC
                                                                ;TEST NUMBER
                                                                ;GOOD ECC1
                                                                ;GOOD EC2C
                                                                ;WRITTEN ECC1
                                                                ;WRITTEN ECC2
                                                                ;DATA USED

004134  070142                        DT34                    ;$ERRPC,TSTNM,GECC1,GECC2,WECC1,WECC2,DISK

004136  070436                        DF34                    ;0,0,0,0,0,0,0

                              ;ERROR 32

004140  060544                        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

004142  066707                        DH35                    ;PC
                                                                ;TEST NUMBER
                                                                ;GOOD ECC1
                                                                ;GOOD ECC2
                                                                ;PATTERN REGISTER
                                                                ;RHER1

004144  070162                        DT35                    ;$ERRPC,TSTNM,GECC1,GECC2,EC2,ER1
```

```
                                        DF35                    ;0,0,0,0,0,0

                                ;ERROR 33

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

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

                                        DF36                    ;0,0,0,0,0,0

                                ;ERROR 34

                                        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
                                        DH36                    ;PC
                                                                ;PC OF JSR
                                                                ;TEST NUMBER
                                                                ;RHMR
                                                                ;POSITION REG.
                                                                ;PATTERN REGISTER

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

                                        DF36                    ;0,0,0,0,0,0

                                ;ERROR 35

                                        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

                                        DH37                    ;PC
                                                                ;TEST NUMBER
                                                                ;ECC POSITION
                                                                ;GOOD POSITION
                                                                ;GOOD ECC1
                                                                ;GOOD ECC2
                                                                ;ECC PATTERN
                                                                ;DATA ENVELOPE
                                                                ;N-CODE ZEROS

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

                                        DF37                    ;0,0,0,0,0,0,0,0,0

                                ;ERROR 36
```

```
                                                                ;ON A READ COMMAND WITH NON CORRECTABLE
                                                                ;ERROR INSERTED DCK AND ECH SHOULD BE SET
                                                                ;PC
                                                                ;TEST NUMBER
                                                                ;GOOD ECC1
                                                                ;GOOD ECC2
                                                                ;PATTERN REGISTER
                                                                ;POSITION REGISTER
                                                                ;RHER1

                                                                ;$ERRPC,TSTNM,GECC1,GECC2,EC2,EC1,ER1

                                                                ;0,0,0,0,0,0,0

                                ;ERROR 37

                                                                ;PGE ERROR
                                                                ;PC
                                                                ;TEST NO
                                                                ;FAILING REG. ADDR
                                                                ;CONTENTS OF RHCS1
                                                                ;CONTENTS OF RHCS2
                                                                ;CONTENTS OF RHDS1
                                                                ;CONTENTS OF RHER1

                                                                ;$ERRPC,$TSTNM,$BDADR,CS1,CS2,DS1,ER1

                                                                ;0,0,0,0,0,0

                                ;ERROR 40

                                                                ;RHWC DID NOT = 0 AFTER A READ OR
                                                                ;WRITE HEADER AND DATA

                                                                ;PC
                                                                ;TEST NO
                                                                ;CONTENTS OF RHWC

                                                                ;$ERRPC,TSTNM,$BDDAT

                                                                ;0,0,0
```

ERROR POINTER TABLE

```
                                        ;THIS ROUTINE HANDLES UNEXPECTED TIMEOUTS
                                        
004230  011600                  BADTMO: MOV     (SP),R0         ;SAVE PC WHERE THE TIME OUT OCCURED
004232  005740                          TST     -(R0)           ;ADJUST PC -2
004234  022626                          CMP     (SP)+,(SP)+     ;RESTORE STACK POINTER
004236  104401  004244                  TYPE    ,65$            ;;TYPE ASCIZ STRING
004242  000417                          BR      64$             ;;GET OVER THE ASCIZ
                                ;;65$:  .ASCIZ  <CRLF>/UNEXPECTED BUS TIMEOUT, PC=/
        0043C2                  64$:
004302  010046                          MOV     R0,-(SP)        ;SETUP FOR TYPING OUT PC
004304  104402                          TYPOC
004306  000240                          NOP                     ;PUT 'HALT(0)' INSTRUCTION HERE IF YOU WISH
                                                                ;TO STOP ON UNEXPECTED TIMEOUT.
                                        
                                        .SBTTL  START OF PROGRAM
                                        
004310  005037  001402          BEGIN:  CLR     SELECT          ;DO NOT SELECT UNIT, NORMAL RUN
004314  000403                          BR      START
                                        
004316  012737  177777  001402  BEGIN2: MOV     #-1,SELECT      ;SELECT UNIT
                                        
004324  005227  000000          START:  INC     #0              ;TTY LOOP, WAIT FOR INCREMENT
004330  001375                          BNE     .-4             ;OF WORD
004332  000005                          RESET                   ;RESET THE WORLD
                                        
                                        .SBTTL  INITIALIZE THE COMMON TAGS
                                        ;;CLEAR THE COMMON TAGS ($CMTAG) AREA
004334  012706  001100                  MOV     #$CMTAG,R6      ;;FIRST LOCATION TO BE CLEARED
004340  005026                          CLR     (R6)+           ;;CLEAR MEMORY LOCATION
004342  022706  001140                  CMP     #SWR,R6         ;;DONE?
004346  001374                          BNE     .-6             ;;LOOP BACK IF NO
004350  012706  001100                  MOV     #STACK,SP       ;;SETUP THE STACK POINTER
                                ;;INITIALIZE A FEW VECTORS
004354  012737  053020  000020          MOV     #$SCOPE,@#IOTVEC ;;IOT VECTOR FOR SCOPE ROUTINE
004362  012737  000340  000022          MOV     #340,@#IOTVEC+2 ;;LEVEL 7
004370  012737  055470  000030          MOV     #$ERROR,@#EMTVEC ;;EMT VECTOR FOR ERROR ROUTINE
004376  012737  000340  000032          MOV     #340,@#EMTVEC+2 ;;LEVEL 7
004404  012737  056620  000034          MOV     #$TRAP,@#TRAPVEC ;;TRAP VECTOR FOR TRAP CALLS
004412  012737  000340  000036          MOV     #340,@#TRAPVEC+2;LEVEL 7
004420  012737  056710  000024          MOV     #$PWRDN,@#PWRVEC ;;POWER FAILURE VECTOR
004426  012737  000340  000026          MOV     #340,@#PWRVEC+2 ;;LEVEL 7
004434  005037  001212                  CLR     $TIMES          ;;INITIALIZE NUMBER OF ITERATIONS
004440  005037  001214                  CLR     $ESCAPE         ;;CLEAR THE ESCAPE ON ERROR ADDRESS
004444  112737  000001  001115          MOVB    #1,$ERMAX       ;;ALLOW ONE ERROR PER TEST
004452  012737  004452  001106          MOV     #.,$LPADR       ;;INITIALIZE THE LOOP ADDRESS FOR SCOPE
004460  012737  004460  001110          MOV     #.,$LPERR       ;;SETUP THE ERROR LOOP ADDRESS
                                ;;SIZE FOR A HARDWARE SWITCH REGISTER. IF NOT FOUND OR IT IS
                                ;;EQUAL TO A "-1", SETUP FOR A SOFTWARE SWITCH REGISTER.
004466  013746  000004                  MOV     @#ERRVEC,-(SP)  ;;SAVE ERROR VECTOR
004472  012737  004526  000004          MOV     #64$,@#ERRVEC   ;;SET UP ERROR VECTOR
004500  012737  177570  001140          MOV     #DSWR,SWR       ;;SETUP FOR A HARDWARE SWICH REGISTER
004506  012737  177570  001142          MOV     #DDISP,DISPLAY  ;;AND A HARDWARE DISPLAY REGISTER
004514  022777  177777  174416          CMP     #-1,@SWR        ;;TRY TO REFERENCE HARDWARE SWR
004522  001012                          BNE     66$             ;;BRANCH IF NO TIMEOUT TRAP OCCURRED
                                                                ;;AND  THE HARDWARE SWR IS NOT = -1
004524  000403                          BR      65$             ;;BRANCH IF NO TIMEOUT
004526  012716  004534          64$:    MOV     #65$,(SP)       ;;SET UP FOR TRAP RETURN
```

INITIALIZE THE COMMON TAGS

```
004532  000002                          RTI
004534  012737  000176  001140  65$:    MOV     #SWREG,SWR      ;;POINT TO SOFTWARE SWR
004542  012737  000174  001142          MOV     #DISPREG,DISPLAY
004550  012637  000004          66$:    MOV     (SP)+,@#ERRVEC  ;;RESTORE ERROR VECTOR

24                                      ;SETUP "TIMEOUT" TRAP VECTOR FOR UNEXPECTED BUS TIMEOUTS
25 004554  012737  004230  000004        MOV     #BADTMO,ERRVEC  ;SETUP FOR UNEXPECTED TIMEOUT
26 004562  012737  000300  000006        MOV     #PR6,ERRVEC+2   ;LEVEL 6
27
31                                      .SBTTL  TYPE PROGRAM NAME
                                        ;;TYPE THE NAME OF THE PROGRAM IF FIRST PASS
004570  005227  177777                    INC     #-1             ;;FIRST TIME?
004574  001030                          BNE     67$             ;;BRANCH IF NO
004576  104401  004604                    TYPE    ,68$            ;;TYPE ASCIZ STRING
004602  000425                          BR      67$             ;;GET OVER THE ASCIZ
                                        ;;68$:  .ASCIZ  <CRLF>@CZRJGE0 - RP04/5/6 DISKLESS TEST, PT 1@<CRLF>
004656                                  67$:
                                        .SBTTL  GET VALUE FOR SOFTWARE SWITCH REGISTER
004656  005737  000042                    TST     @#42            ;;ARE WE RUNNING UNDER XXDP/ACT?
004662  001006                          BNE     69$             ;;BRANCH IF YES
004664  023727  001140  000176            CMP     SWR,#SWREG      ;;SOFTWARE SWITCH REG SELECTED?
004672  001005                          BNE     70$             ;;BRANCH IF NO
004674  104406                          GTSWR                   ;;GET SOFT-SWR SETTINGS
004676  000403                          BR      70$
004700  112737  000001  001134  69$:    MOVB    #1,$AUTOB       ;;SET AUTO-MODE INDICATOR
004706                                  70$:
32 004706  005227  177777                INC     #-1             ;FIRST TIME THRU ?
33 004712  001002                       BNE     1$              ;BR IF NO
34 004714  104401  062040                TYPE    ,PREAMB         ;TYPE PRE-AMBLE MESSAGE
35 004720  005037  177776          1$:   CLR     PS              ;SET PROCESSOR STATUS TO 0
36 004724  012777  052740  174274        MOV     #RPVECT,@RPVEC  ;THIS IS FOR UNTIMELY DRIVE INTERRUPTS
37 004732  004737  054216                JSR     PC,$TKINT       ;INITIALIZE THE TTY KEYBOARD
38
39 004736  032777  010000  174174  RH70CK: BIT   #SW12,@SWR      ;LOOK TO SEE IF USING RH70
40 004744  001403                       BEQ     1$              ;IF SW12 = 0, SKIP NEXT
41 004746  012737  000001  001436        MOV     #1,RH70         ;IF SW12 = 1, CU IS AN RH70
42
43 004754  005737  001402          1$:   TST     SELECT          ;200 START?
44 004760  001434                       BEQ     TST1            ;SKIP NEXT IF STARTING FROM 200
45 004762  104401  004770                TYPE    ,65$            ;;TYPE ASCIZ STRING
   004766  000422                       BR      64$             ;;GET OVER THE ASCIZ
                                        ;;65$:  .ASCIZ  <CRLF>/SELECT UNIT NUMBER TO BE TESTED ?/
   005034                               64$:
46 005034  104412                       RDOCT
47 005036  042716  177770                BIC     #177770,(SP)    ;ONLY KEEP LAST 3 BITS
48 005042  011637  001374                MOV     (SP),UNIT       ;SAVE UNIT TO BE TESTED
49 005046  012637  001404                MOV     (SP)+,UNITSL    ;SAVE UNIT TO BE TESTED
```

GET VALUE FOR SOFTWARE SWITCH REGISTER

```
                                ;;*********************************************************************
                                ;*TEST 1         REFERENCE EACH REGISTER
                                ;*REFERENCE EACH REGISTER BY A MOVE INSTRUCTION
                                ;;*********************************************************************
005052  000004                TST1:   SCOPE
005054  012737  000001  001212        MOV     #1,$TIMES       ;;DO 1 ITERATION
005062  012706  001100        MOV     #STACK,SP       ;SET UP STACK POINTER
005066  012737  000001  001432        MOV     #1,TSTNM        ;MOVE #1 TO TEST NUMBER

005074  012737  055504  000030        MOV     #REGSA1,EMTVEC  ;ERROR VECTOR SO THAT NO REGISTERS ARE SAVED
005102  013746  000004        MOV     ERRVEC,-(SP)    ;;PUSH ERRVEC ON STACK
005106  012737  005140  000004        MOV     #2$,ERRVEC      ;SETUP FOR BUS TIMEOUT
005114  012700  000024        MOV     #24,R0          ;THERE ARE 24 REG TO TEST
005120  012701  001230        MOV     #RHDB,R1        ;R1 NOW HAS ADDR OF ADDR OF FIRST REG.
005124  013102                1$:     MOV     @(R1)+,R2       ;READ HARDWARE REG.
005126  005300                DEC     R0              ;COUNT DOWN
005130  001375                BNE     1$              ;BRANCH IF 24 NOT DONE
005132  012637  000004        MOV     (SP)+,ERRVEC    ;;POP STACK INTO ERRVEC
005136  000455                BR      5$              ;BRANCH IF 24 DONE

005140  012716  005146        2$:     MOV     #3$,(SP)        ;SETUP RETURN ADDRESS
005144  000002                RTI
005146                        3$:
005146  012637  000004        MOV     (SP)+,ERRVEC    ;;POP STACK INTO ERRVEC
005152  016137  177776  001200        MOV     -2(R1),$TMP1    ;STORE FAILING REG ADDR
005160  104015                EMT     15
005162  032777  020000  173750        BIT     #SW13,@SWR      ;INHIBIT ERROR PRINTOUT ?
005170  001036                BNE     4$              ;BRANCH IF YES
005172  104401  005200        TYPE    ,65$            ;;TYPE ASCIZ STRING
005176  000427                BR      64$             ;;GET OVER THE ASCIZ
                                ;;65$:  .ASCIZ  <CRLF>/TO CHANGE BASE ADDRESS, RESTART AT ADDRESS /
005256                        64$:
005256  012746  000204        MOV     #204,-(SP)      ;GET READY TO TYPE STARTING ADDRESS
                                                        ;OF ''CHANGE OF BASE ADDRESS'' ROUTINE
005262  104402                TYPOC
005264  000000                HALT                    ;FORCE THE RESTART!
005266  000137  040164        4$:     JMP     $EOP            ;GO TO END OF PROGRAM

005272                        5$:
005272  013746  000004        MOV     ERRVEC,-(SP)    ;;PUSH ERRVEC ON STACK
005276  012737  005354  000004        MOV     #6$,ERRVEC      ;INITIALIZE VECTOR
005304  005037  001436        CLR     RH70            ;INIT RH INDICATOR ++ C.W
005310  005777  173764        TST     @RHBAE          ;ADDRESS RPBAE (RH11/RH70?)
005314  005237  001436        INC     RH70            ;FOUND AN RH70-SET MASK
005320  104401  005326        TYPE    ,67$            ;;TYPE ASCIZ STRING
005324  000412                BR      66$             ;;GET OVER THE ASCIZ
                                ;;67$:  .ASCIZ  <CRLF><LF>/RH70 CONTROLLER /
005352                        66$:
005352  000420                BR      8$

005354  012716  005362        6$:     MOV     #7$,(SP)        ;SETUP RETURN ADDRESS
005360  000002                RTI
005362                        7$:
005362  104401  005370        TYPE    ,69$            ;;TYPE ASCIZ STRING
005366  000412                BR      68$             ;;GET OVER THE ASCIZ
                                ;;69$:  .ASCIZ  <CRLF><LF>/RH11 CONTROLLER /
```

```
                                                68$:
    005414
 45 005414  012737  055470  000030  8$:       MOV     #$ERROR,EMTVEC  ;RESTORE ERROR VECTOR SO REGISTERS ARE SAVED
 46 005422  012637  000004                     MOV     (SP)+,ERRVEC    ;;POP STACK INTO ERRVEC
 47
 48                                             ;FIND THE SILO SIZE
 49
 50 005426  004737  041460                     JSR     PC,CLDISK       ;CONTROLLER CLEAR
 51 005432  005037  001440                     CLR     SILOSZ          ;CLEAR SILO COUNTER
 52 005436  013777  001440  173564  9$:       MOV     SILOSZ,@RHDB    ;LOAD SILO
 53 005444  005237  001440                     INC     SILOSZ          ;KEEP COUNT
 54 005450  032777  000100  173560             BIT     #IR,@RHCS2      ;IS THE SILO FULL?
 55 005456  001367                             BNE     9$              ;BRANCH IF NO
 56 005460  012737  000412  036150             MOV     #266.,VAR1+2    ;VAR1 IN TEST 116
 57 005466  163737  001440  036150             SUB     SILOSZ,VAR1+2
 58 005474  005437  036150                     NEG     VAR1+2
 59 005500  012737  001510  036156             MOV     #WRFROM,VAR2+2  ;VAR2 IN TEST 116
 60 005506  063737  001440  036156             ADD     SILOSZ,VAR2+2
 61 005514  063737  001440  036156             ADD     SILOSZ,VAR2+2
 62 005522  062737  001000  036156             ADD     #512.,VAR2+2
 63 005530  004737  041460                     JSR     PC,CLDISK       ;CONTROLLER CLEAR
 64
 69
                                                ;;*****************************************************************
                                                ;*TEST 2        RHCS2-CONTROL AND STATUS 2
                                                ;*THIS PARTIALLY TESTS RHCS2 TO ENABLE DETERMINATION
                                                ;*OF THE NUMBER OF DRIVES PRESENT
                                                ;;*****************************************************************
    005534  000004                              TST2:   SCOPE
    005536  012737  000001  001212              MOV     #1,$TIMES       ;;DO 1 ITERATION
 70 005544  012706  001100                      MOV     #STACK,SP       ;RESET STACK
 71 005550  012737  000002  001432              MOV     #2,TSTNM        ;MOVE #2 TO TEST NUMBER
 72
                                                ;CHECK TO SEE IF THE PROGRAM IS RUNNING WITH AN RH70

    005556  005737  001436                      TST     RH70            ;TEST FLAG FOR RH70 CONTROLLER
    005562  001402                              BEQ     64$             ;IF FLAG = 1, THIS TEST IS SKIPPED
    005564  000137  005620                      JMP     TST3            ;JUMP TO NEXT TEST
    005570                                      64$:                    ;IF FLAG = 0, DO THIS TEST
 73
 74 005570  013737  007610  005610              MOV     PRCS2+12,UN
 75 005576  013737  001236  005612              MOV     RHCS2,UN+2
 76 005604  004537  041132                      JSR     R5,BITST        ;TEST BITS IN REGISTER
 77 005610  000000                              UN:     .WORD   0       ;ONLY THESE BITS TESTED FOR READ/WRITE
 78 005612  000000                              .WORD   0               ;ADDRESS OF REG. BEING TESTED
 79 005614  104001                              EMT     1
 80 005616  000207                              RTS     PC              ;RETURN TO BLT3 ROUTINE
 81
 82
                                                ;;*****************************************************************
                                                ;*TEST 3        PARTIAL TEST OF RHAS FOR UNIT NUMBERS PRESENT
                                                ;;*****************************************************************
    005620  000004                              TST3:   SCOPE
    005622  012737  000001  001212              MOV     #1,$TIMES       ;;DO 1 ITERATION
 83 005630  012737  000003  001432              MOV     #3,TSTNM        ;MOVE #3 TO TEST NUMBER
 84
 85 005636  013701  001256                      MOV     RHAS,R1         ;R1 HAS ADDRESS OF RHAS
 86 005642  012711  177777                      MOV     #-1,@R1         ;THIS CLEARS RHAS (SURPRISED!)
 87 005646  011137  001126                      MOV     @R1,$BDDAT      ;TEST DATA
 88 005652  105737  001126                      TSTB    $BDDAT
```

```
                                        BEQ     TST4            ;BRANCH IF GOOD
                                        CLR     $GDDAT          ;GOOD DATA
                                        MOV     R1,REGADR       ;FAILING REG. RHAS
                                        EMT     1
                                                                ;BY WRITING ONES IN


                                ;;**********************************************************************
                                ;*TEST 4         TEST FOR DRIVES PRESENT USING RHAS AND RHCS2
                                ;;**********************************************************************
005672  000004                  TST4:   SCOPE
005674  012737  000001  001212          MOV     #1,$TIMES       ;;DO 1 ITERATION
                                        RESET                   ;START WITH AN INIT
                                        MOV     #4,TSTNM        ;MOVE #4 TO TEST NUMBER
                                        JSR     PC,$TKINT       ;INITILIZE TTY KEYBOARD
                                        BIT     #SW13,@SWR      ;INHIBIT ERROR TYPEOUT?
                                        BNE     4$              ;BRANCH IF YES
                                        TYPE    ,65$            ;;TYPE ASCIZ STRING
005732  000421                          BR      64$             ;;GET OVER THE ASCIZ
                                ;;65$:  .ASCIZ  <CRLF>/LOOKING AT RHAS - DRIVES PRESENT/
005776                          64$:
                                4$:     MOV     RHAS,R1         ;LOAD R1 WITH ADDR. OF RHAS
                                        MOV     RHCS2,R2        ;LOAD R2 WITH ADDR. OF RHCS2
                                        CLR     @R2             ;CLEAR RHCS2 (ADDRESS UNIT #0)
                                        MOV     #8.,R0          ;INITIALIZE DRIVE COUNTER
                                        MOV     RHER1,R4        ;LOAD R4 WITH ADDR. OF RHER1
                                1$:     MOV     #-1,@R4         ;MOVE ERRORS INTO RHER1 OF UNIT ADDRESSED
                                        INC     @R2             ;INCREMENT UNIT NO. (RHCS2)
                                        DEC     R0              ;COUNT DOWN DRIVE COUNTER
                                        BNE     1$              ;TEST AND DO NEXT UNIT IF 8 NOT DONE

                                        MOVB    @R1,TOTALAT     ;SAVE ALL RESULTING ATTENTION BITS
                                                                ;(USED IN DRIVE CLEAR TEST)
                                        CLRB    TOTALAT+1       ;CLEAR UPPER BYTE
                                        TSTB    @R1             ;TEST RHAS FOR ANY DRIVES PRESENT
                                        BEQ     2$              ;NONE RESPONDING - TYPE THE MESSAGE
                                        JMP     XE2             ;SOME THERE - GO FILL ''UNITS'' TABLE

                                2$:     BIT     #SW13,@SWR      ;INHIBIT ERROR TYPE OUT?
                                        BEQ     3$              ;TYPE ''NO DRIVES'' MESSAGE IF NO
                                        JMP     SELTST          ;CHECK FOR SELECTED UNIT START AND LOAD
                                                                ;''UNITS'' TABLE WITH DESIRED DRIVE IF SO

                                3$:     TYPE    ,NDRVAS         ;TYPE 'NO DRIVES PRESENT IN RHAS'
                                        JMP     $EOP            ;GO OUT

                                ;SET UP DRIVES PRESENT TABLE
006076                          XE2:

                                2$:     MOV     #8.,R0          ;LOAD ''UNITS'' TABLE COUNTER
                                        MOV     #UNITS,R3       ;LOAD ''UNITS'' TABLE POINTER
                                3$:     MOV     #-1,(R3)+       ;PRESET 1ST TABLE BLOCK TO ALL ONES
                                        DEC     R0              ;COUNT DOWN
                                        BNE     3$              ;PRESET NEXT BLOCK IF 8 NOT DONE

                                10$:    MOV     #UNITS,R3       ;RELOAD THE TABLE POINTER
                                        CLR     R5              ;INITIALIZE UNIT NO. TO 0
                                        CLR     NOUNIT  ;NO. OF UNITS PRESENT
```

```
                                                MOV     #8.,R0          ;RELOAD THE TABLE COUNTER
                                                MOV     @R1,$TMP0       ;ADDR OF RHAS INTO TEMPORARY STORAGE
                                        4$:     ROR     $TMP0           ;SET CARRY IF 0 BIT = 1 (UNIT ATTEN.)
                                                BCC     5$              ;CHECK NEXT UNIT IF ONE NOT IN BIT 0

                                        11$:    MOV     R5,@RHCS2       ;INSERT UNIT NO. INTO RHCS2 UNIT ADDR.
                                                CMP     #24020,@RHDT    ;READ RHDT - IS IT A DUAL PORT RP04 ?
                                                BEQ     6$              ;YES...TYPE THE UNIT NO.
                                                CMP     #20020,@RHDT    ;READ RHDT - IS IT A SINGLE PORT RP04 ?
                                                BEQ     6$              ;YES...TYPE THE UNIT NO.

                                                CMP     #24021,@RHDT    ;DUAL PORT RP05 ?
                                                BEQ     6$              ;TYPE UNIT NO. IF SO
                                                CMP     #20021,@RHDT    ;SINGLE PORT RP05 ?
                                                BEQ     6$              ;TYPE NO. IF SO

                                                CMP     #24022,@RHDT    ;READ RHDT - IS IT A DUAL PORT RP06 ?
                                                BEQ     6$              ;YES...TYPE THE UNIT NO.
                                                CMP     #20022,@RHDT    ;READ RHDT - IS IT A SINGLE PORT RP06 ?
                                                BEQ     6$              ;YES...TYPE THE UNIT NO.

                                                TYPE    ,65$            ;;TYPE ASCIZ STRING
                                                BR      64$             ;;GET OVER THE ASCIZ
                                        ;;65$:  .ASCIZ  <CRLF>/UNIT NUMBER /
                                        64$:
                                                MOV     R5,-(SP)        ;PUT THE UNIT NUMBER ON STACK
                                                TYPDS                   ;TYPE IT
                                                TYPE    ,67$            ;;TYPE ASCIZ STRING
                                                BR      66$             ;;GET OVER THE ASCIZ
                                        ;;67$:  .ASCIZ  /, RHDT = /
                                        66$:
                                                MOV     @RHDT,-(SP)     ;PUT RHDT ON THE STACK
                                                TYPOC                   ;TYPE IT
                                                TYPE    ,69$            ;;TYPE ASCIZ STRING
                                                BR      68$             ;;GET OVER THE ASCIZ
                                        ;;69$:  .ASCIZ  ? - NOT AN RP04/RP05/RP06 DEVICE?
                                        68$:
                                                BR      5$              ;UNIT NOT AN RP04/RP05/RP06 SO TEST NEXT ONE

                                        6$:     MOV     R5,(R3)+        ;LOAD TABLE POSITION AND INCR IT
                                                TYPE    ,$CRLF          ;CRLF
                                                MOV     R5,-(SP)        ;PUT UNIT NO. ON THE STACK
                                                TYPDS                   ;TYPE THE UNIT NO.
                                                INC     NOUNIT          ;INCR THE TOTAL NO. OF UNITS

                                        5$:     INC     R5              ;'RHCS2' UNIT ADDRESS
                                                DEC     R0              ;DRIVE COUNTER DOWN ONE
                                                BNE     4$              ;TEST AND DO NEXT UNIT IF 8 NOT DONE

                                        12$:    MOV     UNITS,UNIT      ;SET UNIT NO. TO FIRST ONE FOUND/OR 0
                                                MOV     NOUNIT,NUNIT    ;SAVE NO. OF UNITS
                                                DEC     NUNIT           ;IF NUNIT = 0 THEN ONLY ONE UNIT
                                                                        ;IF NUNIT > 0 THEN MORE THAN ONE UNIT

                                        SELTST: TST     SELECT          ;STARTING ADDRESS 200 ?
                                                BEQ     TST5            ;BRANCH IF STARTING FROM 200
                                                MOV     UNITSL,UNIT     ;CHANGE UNIT NUMBER TO SELECTED ONE
```

```
                                ;;*****************************************************************
                                ;*TEST 5         TYPE SERIAL NUMBER AND DRIVE TYPE
                                ;*SET APPROPRIATE ATTENTION BIT OF UNIT UNDER TEST IN
                                ;*'ATTENT' AND TYPE THE UNIT UNDER TEST
                                ;*
                                ;*READ SERIAL NUMBER REGISTER AND DRIVE TYPE REGISTER,
                                ;*TYPE THEM OUT AND PROCEED
                                ;*
                                ;*TO LOOP HERE SET SWITCH 8 AND THIS TEST NO. AND RESTART
                                ;;*****************************************************************
006440  000004                  TST5:   SCOPE
006442  012737  000001  001212          MOV     #1,$TIMES       ;;DO 1 ITERATION
006450  012737  007156  001106          MOV     #1$,$LPADR      ;;SET SCOPE LOOP ADDRESS
198 006456  012737  000005  001432      MOV     #5,TSTNM        ;MOVE #5 TO TEST NUMBER
199 006464  004737  041460              JSR     PC,CLDISK       ;FILL UNIT NO.
200 006470  005037  001416              CLR     ATTENT  ;CLEAR
201
202                             ;TEST FOR UNIT #0
203
204 006474  005737  001374              TST     UNIT            ;IS UNIT #0 NEXT IN THE UNITS TABLE ?
205 006500  001022                      BNE     10$             ;IF NOT, TEST THIS UNIT
206 006502  012700  000041              MOV     #41,R0          ;IF SO, CHECK THE LOAD MEDIA LOCATION
207 006506  122710  000011              CMPB    #11,(R0)        ;WAS IT AN RP04/5/6 ?
208 006512  001015                      BNE     10$             ;NO...GO AHEAD WITH TESTING UNIT #0
209 006514  005737  001402              TST     SELECT          ;WAS UNIT #0 SELECTED ?
210                                                             ;(IE. WAS IT A 210 START ?)
211 006520  001012                      BNE     10$             ;IF SO...TEST IT
212
213                             ;INCREMENT THE UNITS TABLE TO NEXT DRIVE (IF ANY)
214                             ;& DECREMENT THE 'NOUNITS' PRESENT (TO BE TESTED)
215
216 006522  012700  001354              MOV     #UNITS,R0       ;LOAD THE UNITS TABLE POINTER
217 006526  005720                      TST     (R0)+           ;SELECT THE NEXT UNIT IN THE TABLE
218                                                             ;(DOUBLE INCREMENT THE POINTER, R0)
219 006530  022710  177777              CMP     #-1,(R0)        ;IS THERE ANOTHER TABLE ENTRY PRESENT ?
220 006534  001404                      BEQ     10$             ;IF NOT (LOC = -1)...MUST USE UNIT #0
221 006536  011037  001374              MOV     (R0),UNIT       ;SET UP TO BE THE UNIT UNDER TEST
222 006542  005337  001376              DEC     NOUNITS ;DECREMENT BECAUSE UNIT # 0 WONT'T BE TESTED
223 006546  013700  001374      10$:    MOV     UNIT,R0 ;R0 CONTAINS UNIT NO.
224
225                             ;SET UP THE PROPER DEVICE TYPE FLAG
226
227 006552  010077  172460              MOV     R0,@RHCS2       ;SET UP UNIT ADDRESS
228 006556  005037  001434              CLR     RP06            ;CLEAR RP06 DEVICE TYPE FLAG
229 006562  022777  024022  172474      CMP     #24022,@RHDT    ;DUAL PORT RP06 ?
230 006570  001405                      BEQ     2$              ;YES...SET THE FLAG
231 006572  022777  020022  172464      CMP     #20022,@RHDT    ;SINGLE PORT RP06 ?
232 006600  001401                      BEQ     2$              ;YES..SET FLAG
233 006602  000403                      BR      3$              ;NO...DON'T SET RP06 FLAG
234 006604  012737  177777  001434  2$: MOV     #-1,RP06        ;SET IT
235
236                             ;ASSUME THE NEXT UNIT IS AN RP04
237
238 006612  116037  003620  001416  3$: MOVB    ATABLE(R0),ATTENT ;SET APPROPRIATE ATTENTION BIT
239 006620  104401  006626              TYPE    ,65$            ;;TYPE ASCIZ STRING
    006624  000413                      BR      64$             ;;GET OVER THE ASCIZ
```

```
                                        ;;65$:  .ASCIZ  <CRLF>/TESTING DRIVE NUMBER/
     006654                             64$:
 240 006654  013746  001374                     MOV     UNIT,-(SP)      ;UNIT NO. TO STACK
 241 006660  104405                             TYPDS                   ;TYPE  DRIVE NO.
 242 006662  104401  006670                     TYPE    ,67$            ;;TYPE ASCIZ STRING
     006666  000410                             BR      66$             ;;GET OVER THE ASCIZ
                                        ;;67$:  .ASCIZ  <CRLF>/SERIAL NO. = /
     006710                             66$:
 243 006710  017746  172352                     MOV     @RHSN,-(SP)     ;;SAVE @RHSN FOR TYPEOUT
     006714  104402                             TYPOC                   ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
 244 006716  104401  006724                     TYPE    ,69$            ;;TYPE ASCIZ STRING
     006722  000410                             BR      68$             ;;GET OVER THE ASCIZ
                                        ;;69$:  .ASCIZ  <CRLF>/DRIVE TYPE = /
     006744                             68$:
 245 006744  017746  172314                     MOV     @RHDT,-(SP)     ;;SAVE @RHDT FOR TYPEOUT
     006750  104402                             TYPOC                   ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
 246
 247 006752  022777  024020  172304             CMP     #24020,@RHDT    ;DUAL PORT RP04 ?
 248 006760  001425                             BEQ     4$              ;TYPE ASCII MSG OUT
 249 006762  022777  020020  172274             CMP     #20020,@RHDT    ;SINGLE PORT RP04 ?
 250 006770  001421                             BEQ     4$              ;TYPE THE MESSAGE
 251
 252 006772  022777  024021  172264             CMP     #24021,@RHDT    ;DUAL PORT RP05 ?
 253 007000  001451                             BEQ     6$              ;TYPE MSG
 254 007002  022777  020021  172254             CMP     #20021,@RHDT    ;SINGLE PORT RP05 ?
 255 007010  001445                             BEQ     6$              ;TYPE MSG
 256
 257 007012  022777  024022  172244             CMP     #24022,@RHDT    ;DUAL PORT RP06 ?
 258 007020  001423                             BEQ     5$              ;TYPE MSG
 259 007022  022777  020022  172234             CMP     #20022,@RHDT    ;SINGLE PORT RP06 ?
 260 007030  001417                             BEQ     5$              ;TYPE MSG
 261 007032  000451                             BR      1$              ;DRIVE IS NOT AN RP04/RP05/RP06  - SO
 262                                                                    ;DON'T TYPE THE ASCII MESSAGE
 263                                                                    ;-SHOULD NEVER HAPPEN AT THIS POINT
 264                                                                    ;UNLESS DRIVE GOT SICK WHILE TESTING
 265                                                                    ;WAS IN PROGRESS
 266
     007034                             4$:
 267 007034  104401  007042                     TYPE    ,71$            ;;TYPE ASCIZ STRING
     007040  000412                             BR      70$             ;;GET OVER THE ASCIZ
                                        ;;71$:  .ASCIZ  <CRLF>/DRIVE IS AN RP04/<CRLF>
     007066                             70$:
 268 007066  000433                             BR      1$              ;SKIP NEXT MESSAGE
 269 007070                             5$:
     007070  104401  007076                     TYPE    ,73$            ;;TYPE ASCIZ STRING
     007074  000412                             BR      72$             ;;GET OVER THE ASCIZ
                                        ;;73$:  .ASCIZ  <CRLF>/DRIVE IS AN RP06/<CRLF>
     007122                             72$:
 270 007122  000415                             BR      1$              ;SKIP NEXT
 271 007124                             6$:
     007124  104401  007132                     TYPE    ,75$            ;;TYPE ASCIZ STRING
     007130  000412                             BR      74$             ;;GET OVER THE ASCIZ
                                        ;;75$:  .ASCIZ  <CRLF>/DRIVE IS AN RP05/<CRLF>
     007156                             74$:
 272
 273 007156  005777  172104             1$:     TST     @RHSN           ;READ SERIAL NO. AND DRIVE TYPE
 274 007162  005777  172076                     TST     @RHDT           ;THESE TWO ARE TO HELP SCOPE LOOPS
```

```
    275 007166  017737  172074  001412          MOV     @RHSN,SAVSN     ;SAVE TO CHECK IF CLR RHCS2 BIT 5 CLEARS ANY BITS
    276 007174  017737  172064  001410          MOV     @RHDT,SAVDT     ;SAVE TO CHECK IF CLR RHCS2 BIT 5 CLEARS ANY BITS
    277
    283                                 ;;*****************************************************************
                                        ;*TEST 6         CHECK MOL TO BE LOW
                                        ;*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
                                        ;;*****************************************************************
        007202  000004                  TST6:   SCOPE
    284 007204  012737  000006  001432          MOV     #6,TSTNM        ;MOVE #6 TO TEST NUMBER
    285 007212  004737  041460          JSR     PC,CLDISK       ;GIVE INITILIZE
    286 007216  032713  010000          BIT     #MOL,@R3        ;CHECK MOL IN RHDS1
    287 007222  001542                  BEQ     TST7            ;BRANCH IF MOL LOW
    288 007224  104401  007232          TYPE    ,65$            ;;TYPE ASCIZ STRING
        007230  000420                  BR      64$             ;;GET OVER THE ASCIZ
                                        ;;65$:  .ASCIZ  <CRLF>/DRIVE IS ON LINE - MOL IS HIGH/
        007272                          64$:
    289 007272  104401  007300          TYPE    ,67$            ;;TYPE ASCIZ STRING
        007276  000423                  BR      66$             ;;GET OVER THE ASCIZ
                                        ;;67$:  .ASCIZ  <CRLF>/HIT STOP ON DRIVE TO GET IT OFF LINE/
        007346                          66$:
    290 007346  104401  007354          TYPE    ,69$            ;;TYPE ASCIZ STRING
        007352  000430                  BR      68$             ;;GET OVER THE ASCIZ
                                        ;;69$:  .ASCIZ  <CRLF>/PROGRAM WILL HANG TESTING MOL TILL MOL IS LOW/
        007434                          68$:
    291 007434  032713  010000          1$:     BIT     #MOL,@R3        ;CHECK MOL IN RHDS1
    292 007440  001375                  BNE     1$              ;BRANCH IF MOL IS HIGH
    293 007442  104401  007450          TYPE    ,71$            ;;TYPE ASCIZ STRING
        007446  000430                  BR      70$             ;;GET OVER THE ASCIZ
                                        ;;71$:  .ASCIZ  <CRLF><LF>/MOL IS NOW LOW. PROGRAM WILL NOW BE EXECUTED/<CRLF>
        007530                          70$:
```

```
                                .SBTTL  REGISTER TESTS
                                ;;*****************************************************************
                                ;*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)
                                ;;*****************************************************************
007530  000004                  TST7:   SCOPE
 9 007532  012737  000007  001432       MOV     #7,TSTNM        ;MOVE #7 TO TEST NUMBER
10
                                ;CHECK TO SEE IF THE PROGRAM IS RUNNING WITH AN RH70

007540  005737  001436                  TST     RH70            ;TEST FLAG FOR RH70 CONTROLLER
007544  001402                          BEQ     64$             ;IF FLAG = 1, THIS TEST IS SKIPPED
007546  000137  007620                  JMP     TST10           ;JUMP TO NEXT TEST
007552                          64$:                            ;IF FLAG = 0, DO THIS TEST
11 007552  004737  041460               JSR     PC,CLDISK       ;GIVE INITIALIZE
12 007556  012706  001100               MOV     #STACK,SP       ;RESET STACK
007562  012737  000007  001432          MOV     #7,TSTNM        ;MOVE #7 TO TEST NUMBER

007570  013737  001236  007612          MOV     RHCS2,PRCS2+14  ;GET REGISTER ADDRESS
007576  013777  001374  171432  PRCS2:  MOV     UNIT,@RHCS2     ;MOVE UNIT NO. UNDER TEST
007604  004537  041132                  JSR     R5,BITST        ;TEST BITS IN REGISTER
007610  020017                          .WORD   20017           ;ONLY THESE BITS ARE TESTED FOR READ/WRITE
007612  176710                          .WORD   176710          ;ADDRESS OF REG. BEING TESTED
007614  104001                          EMT     1
007616  000207                          RTS     PC              ;RETURN TO BLT3 ROUTINE

13
19                              ;;*****************************************************************
                                ;*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)
                                ;;*****************************************************************
007620  000004                  TST10:  SCOPE
20 007622  012706  001100               MOV     #STACK,SP       ;RESET STACK
007626  012737  000010  001432          MOV     #10,TSTNM       ;MOVE #10 TO TEST NUMBER

007634  013737  001240  007656          MOV     RHCS1,PRCS1+14  ;GET REGISTER ADDRESS
007642  013777  001374  171366  PRCS1:  MOV     UNIT,@RHCS2     ;MOVE UNIT NO. UNDER TEST
007650  004537  041132                  JSR     R5,BITST        ;TEST BITS IN REGISTER
007654  001476                          .WORD   1476            ;ONLY THESE BITS ARE TESTED FOR READ/WRITE
007656  176700                          .WORD   176700          ;ADDRESS OF REG. BEING TESTED
007660  104001                          EMT     1
007662  000207                          RTS     PC              ;RETURN TO BLT3 ROUTINE

21
27                              ;;*****************************************************************
                                ;*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
                                ;;*****************************************************************
007664  000004                  TST11:  SCOPE
28 007666  012706  001100               MOV     #STACK,SP       ;RESET STACK
```

```
007672  012737  000011  001432          MOV     #11,TSTNM       ;MOVE #11 TO TEST NUMBER
007700  004737  041460                  JSR     PC,CLDISK       ;INIT AND SET UNIT NUMBER AND DEVICE -
                                                                ;CPU REG. CORRESPONDENCE (R1-R4)

                                ;SET FORCED PARITY ERROR 'PAT'
007704  052777  000020  171324          BIS     #PAT,@RHCS2     ;SET 'PAT' TO INVERT PARITY
                                                                ;GENERATED
007712  005077  171324                  CLR     @RHER1          ;WRITE DCL REGISTER USING BAD CONTROL PARITY

                                ;WITH THIS PARITY ERROR NOTHING WILL BE READ TILL IT IS
                                ;CLEARED

007716  011137  001126                  MOV     @R1,$BDDAT      ;RHCS1 $BDDAT
007722  022737  104200  001126          CMP     #SC!DVA!RDY,$BDDAT ;COMPARE RHCS1 AFTER PARITY
                                                                ;ERROR
007730  001406                          BEQ     1$              ;BRANCH IF SC!DVA!RDY=1
007732  012737  104200  001124          MOV     #SC!DVA!RDY,$GDDAT ;GOOD DATA
007740  010137  041130                  MOV     R1,REGADR       ;REGISTER ADDRESS RHCS1
007744  104001                          EMT     1
                                                                ;WRITING DCL REGISTER RHCS1
                                                                ;DID NOT SET SC!DVA!RDY
                                                                ;WITH 'PAT' BIT HIGH
007746  013746  001374          1$:     MOV     UNIT,-(SP)      ;GET UNIT NUMBER
007752  052716  000120                  BIS     #PAT!IR,(SP)    ;INCLUDE PAT AND IR
007756  012637  001124                  MOV     (SP)+,$GDDAT    ;PUT ON STACK
007762  011237  001126                  MOV     @R2,$BDDAT      ;RHCS2 $BDDAT
007766  023737  001124  001126          CMP     $GDDAT,$BDDAT ;COMPARE RHCS2
007774  001403                          BEQ     2$              ;OK - SC!DVA!RDY ARE HIGH
007776  010237  041130                  MOV     R2,REGADR       ;REGISTER ADDRESS
010002  104001                          EMT     1
                                                                ;SHOW UNIT#!PAT!IR BITS HIGH

010004  011437  001126          2$:     MOV     @R4,$BDDAT      ;RHER1 $BDDAT
010010  022737  000010  001126          CMP     #PAR,$BDDAT     ;ERROR REGISTER RHER1 SHOULD
                                                                ;HAVE 'PAR' SET
010016  001406                          BEQ     3$              ;A - OK, IT DOES
010020  012737  000010  001124          MOV     #PAR,$GDDAT     ;GOOD DATA
010026  010437  041130                  MOV     R4,REGADR       ;FAILING REGISTER RHER1
010032  104001                          EMT     1
                                                                ;SET 'PAR'
010034                          3$:

                                ;;*****************************************************************
                                ;*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)
                                ;;*****************************************************************
010034  000004                  TST12:  SCOPE
010036  012706  001100                  MOV     #STACK,SP       ;RESET STACK
010042  012737  000012  001432          MOV     #12,TSTNM       ;MOVE #12 TO TEST NUMBER

010050  013737  001232  010072          MOV     RHWC,PRWC+14    ;GET REGISTER ADDRESS
010056  013777  001374  171152  PRWC:   MOV     UNIT,@RHCS2     ;MOVE UNIT NO. UNDER TEST
010064  004537  041132                  JSR     R5,BITST        ;TEST BITS IN REGISTER
010070  177777                          .WORD   177777          ;ONLY THESE BITS ARE TESTED FOR READ/WRITE
010072  176702                          .WORD   176702          ;ADDRESS OF REG. BEING TESTED
```

```
010074  104001                          EMT     1
010076  000207                          RTS     PC              ;RETURN TO BLT3 ROUTINE

 78
 84                             ;;*****************************************************************
                                ;*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)
                                ;;*****************************************************************
010100  000004                 TST13:  SCOPE
 85 010102  012706  001100              MOV     #STACK,SP       ;RESET STACK
010106  012737  000013  001432          MOV     #13,TSTNM       ;MOVE #13 TO TEST NUMBER

010114  013737  001234  010136          MOV     RHBA,PRBA+14    ;GET REGISTER ADDRESS
010122  013777  001374  171106  PRBA:   MOV     UNIT,@RHCS2     ;MOVE UNIT NO. UNDER TEST
010130  004537  041132                  JSR     R5,BITST        ;TEST BITS IN REGISTER
010134  177776                          .WORD   177776          ;ONLY THESE BITS ARE TESTED FOR READ/WRITE
010136  176704                          .WORD   176704          ;ADDRESS OF REG. BEING TESTED
010140  104001                          EMT     1
010142  000207                          RTS     PC              ;RETURN TO BLT3 ROUTINE

 86
 92                             ;;*****************************************************************
                                ;*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)
                                ;;*****************************************************************
010144  000004                 TST14:  SCOPE
 93 010146  012706  001100              MOV     #STACK,SP       ;RESET STACK
010152  012737  000014  001432          MOV     #14,TSTNM       ;MOVE #14 TO TEST NUMBER

010160  013737  001242  010202          MOV     RHER1,PRER1+14  ;GET REGISTER ADDRESS
010166  013777  001374  171042  PRER1:  MOV     UNIT,@RHCS2     ;MOVE UNIT NO. UNDER TEST
010174  004537  041132                  JSR     R5,BITST        ;TEST BITS IN REGISTER
010200  177777                          .WORD   177777          ;ONLY THESE BITS ARE TESTED FOR READ/WRITE
010202  176714                          .WORD   176714          ;ADDRESS OF REG. BEING TESTED
010204  104001                          EMT     1
010206  000207                          RTS     PC              ;RETURN TO BLT3 ROUTINE

 94
100                             ;;*****************************************************************
                                ;*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)
                                ;;*****************************************************************
010210  000004                 TST15:  SCOPE
101 010212  012737  000015  001432      MOV     #15,TSTNM       ;MOVE #15 TO TEST NUMBER
102 010220  004737  041460              JSR     PC,CLDISK       ;SET UNIT NUMBER AND INIT
103 010224  013700  001260              MOV     RHMR,R0 ;R0 HAS MAINTENANCE REG. ADR.
104 010230  012701  000001              MOV     #1,R1           ;R1 HAS DATA
105 010234  012702  000005              MOV     #5,R2           ;R2 HAS COUNT OF NUMBER OF BITS
106 010240  012710  000001      1$:     MOV     #DMD,@R0        ;SET DIAGNOSTIC MODE BIT
107 010244  050110                      BIS     R1,@R0          ;SET DATA IN RHMR
108 010246  010146                      MOV     R1,-(SP)        ;SAVE DATA FOR COMPARES
```

```
                                        BIS     #DMD!400,(SP)   ;INCLUDE BIT 0
                                        MOV     (SP),$GDDAT     ;SAVE FOR ERROR PRINTOUT
                                        CMP     (SP)+,@R0       ;COMPARE DATA
                                        BEQ     2$              ;BRANCH IF GOOD
                                        MOV     @R0,$BDDAT      ;BAD DATA
                                        MOV     R0,REGADR       ;FAILING REG. ADR.
                                        EMT     1
                                                                ;FAILED TO SET INDICATED
                                                                ;BITS
                                2$:     CLC                     ;CLEAR CARRY
                                        ROL     R1              ;GET NEXT DATA
                                        BIS     #400,R1         ;SET UNUSED BITS
                                        BIC     #BIT09,R1       ;CLEAR READ ONLY BIT
                                        DEC     R2              ;COUNT
                                        BNE     1$              ;BRANCH IF 5 BITS NOT DONE


                                ;NOW FLOAT A 0

                                        MOV     #435,R1         ;R1 HAS DATA
                                        MOV     #5,R2           ;R2 HAS COUNT BITS
                                3$:     MOV     #DMD,@R0        ;SET DIAGNOSTIC MODE BITS
                                        BIS     R1,@R0          ;SET DATA IN RHMR
                                        CMP     R1,@R0          ;COMPARE DATA
                                        BEQ     4$              ;BRANCH IF GOOD
                                        MOV     R1,$GDDAT       ;GOOD DATA
                                        MOV     @R0,$BDDAT      ;BAD DATA
                                        MOV     R0,REGADR       ;FAILING REG. ADR. RHMR
                                        EMT     1
                                                                ;DOES NOT ALLOW WRITING
                                                                ;ZEROS
                                4$:     SEC                     ;SET CARRY
                                        ROL     R1              ;GET NEXT DATA
                                        BIC     #BIT05!BIT06!BIT07!BIT09,R1  ;CLEAR READ ONLY BIT
                                        BIS     #BIT08,R1       ;SET BIT ZEROED BY ROL
                                        DEC     R2              ;COUNT IF 5 BITS DONE
                                        BNE     3$              ;BRANCH IF INCOMPLETE


                                ;;*****************************************************************
                                ;*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)
                                ;;*****************************************************************
                                TST16:  SCOPE
                                        MOV     #STACK,SP       ;RESET STACK
                                        MOV     #16,TSTNM       ;MOVE #16 TO TEST NUMBER

                                        MOV     RHDST,PRDST+14  ;GET REGISTER ADDRESS
                                PRDST:  MOV     UNIT,@RHCS2     ;MOVE UNIT NO. UNDER TEST
                                        JSR     R5,BITST        ;TEST BITS IN REGISTER
                                        .WORD   17437           ;ONLY THESE BITS ARE TESTED FOR READ/WRITE
                                        .WORD   176706          ;ADDRESS OF REG. BEING TESTED
                                        EMT     1
                                        RTS     PC              ;RETURN TO BLT3 ROUTINE
```

```
160                                     ;;*****************************************************************
                                        ;*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)
                                        ;;*****************************************************************
 010442  000004                         TST17:  SCOPE
161 010444  012706  001100                      MOV     #STACK,SP       ;RESET STACK
 010450  012737  000017  001432                 MOV     #17,TSTNM       ;MOVE #17 TO TEST NUMBER

 010456  013737  001246  010500                 MOV     RHER2,PRER2+14  ;GET REGISTER ADDRESS
 010464  013777  001374  170544  PRER2:         MOV     UNIT,@RHCS2     ;MOVE UNIT NO. UNDER TEST
 010472  004537  041132                         JSR     R5,BITST        ;TEST BITS IN REGISTER
 010476  177777                               .WORD   177777          ;ONLY THESE BITS ARE TESTED FOR READ/WRITE
 010500  176740                               .WORD   176740          ;ADDRESS OF REG. BEING TESTED
 010502  104001                               EMT     1
 010504  000207                               RTS     PC              ;RETURN TO BLT3 ROUTINE

162
168                                     ;;*****************************************************************
                                        ;*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)
                                        ;;*****************************************************************
 010506  000004                         TST20:  SCOPE
169 010510  012706  001100                      MOV     #STACK,SP       ;RESET STACK
 010514  012737  000020  001432                 MOV     #20,TSTNM       ;MOVE #20 TO TEST NUMBER

 010522  013737  001250  010544                 MOV     RHOF,PROF+14    ;GET REGISTER ADDRESS
 010530  013777  001374  170500  PROF:         MOV     UNIT,@RHCS2     ;MOVE UNIT NO. UNDER TEST
 010536  004537  041132                         JSR     R5,BITST        ;TEST BITS IN REGISTER
 010542  016277                               .WORD   16277           ;ONLY THESE BITS ARE TESTED FOR READ/WRITE
 010544  176732                               .WORD   176732          ;ADDRESS OF REG. BEING TESTED
 010546  104001                               EMT     1
 010550  000207                               RTS     PC              ;RETURN TO BLT3 ROUTINE

170
176                                     ;;*****************************************************************
                                        ;*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)
                                        ;;*****************************************************************
 010552  000004                         TST21:  SCOPE
177 010554  012706  001100                      MOV     #STACK,SP       ;RESET STACK
 010560  012737  000021  001432                 MOV     #21,TSTNM       ;MOVE #21 TO TEST NUMBER

 010566  013737  001252  010610                 MOV     RHCA,PRCA+14    ;GET REGISTER ADDRESS
 010574  013777  001374  170434  PRCA:         MOV     UNIT,@RHCS2     ;MOVE UNIT NO. UNDER TEST
 010602  004537  041132                         JSR     R5,BITST        ;TEST BITS IN REGISTER
 010606  001777                               .WORD   1777            ;ONLY THESE BITS ARE TESTED FOR READ/WRITE
 010610  176734                               .WORD   176734          ;ADDRESS OF REG. BEING TESTED
 010612  104001                               EMT     1
 010614  000207                               RTS     PC              ;RETURN TO BLT3 ROUTINE

178
```

```
                                ;;*****************************************************************
                                ;*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)
                                ;;*****************************************************************
010616  000004                  TST22:  SCOPE
185 010620  012706  001100              MOV     #STACK,SP       ;RESET STACK
    010624  012737  000022  001432      MOV     #22,TSTNM       ;MOVE #22 TO TEST NUMBER

    010632  013737  001254  010654      MOV     RHER3,PRER3+14  ;GET REGISTER ADDRESS
    010640  013777  001374  170370  PRER3:  MOV UNIT,@RHCS2     ;MOVE UNIT NO. UNDER TEST
    010646  004537  041132              JSR     R5,BITST        ;TEST BITS IN REGISTER
    010652  177777                      .WORD   177777          ;ONLY THESE BITS ARE TESTED FOR READ/WRITE
    010654  176742                      .WORD   176742          ;ADDRESS OF REG. BEING TESTED
    010656  104001                      EMT     1
    010660  000207                      RTS     PC              ;RETURN TO BLT3 ROUTINE

186
187                             ;;*****************************************************************
188                             ;OF THE TWENTY REGISTERS (4 IN RH11, 16 IN RP04) ONLY 12 ARE
189                             ;CHECKED IN THE ABOVE TESTS
190                             ;TWO ARE ALREADY TESTED (SERIAL NO. AND DRIVE TYPE)
191                             ;THE OTHER 7 WHICH ARE RHDS1, RHLA, RHCC, RHEC1, RHEC1, RHEC2
192                             ;ARE READ ONLY REGISTERS.  ONE OR ZERO CANNOT BE WRITTEN
193                             ;;*****************************************************************
194
201                             ;;*****************************************************************
                                ;*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
                                ;;*****************************************************************
010662  000004                  TST23:  SCOPE
202 010664  012706  001100              MOV     #STACK,SP       ;RESET STACK
203 010670  012737  000023  001432      MOV     #23,TSTNM       ;MOVE #23 TO TEST NUMBER
204
205 010676  004737  041460              JSR     PC,CLDISK       ;HERE IT IS USED TO SETUP HARDWARE/
206                                                             ;CPU REGISTER CORRESPONDENCE
207                                                             ;R1=RHCS1
208                                                             ;R2=RHCS2
209                                                             ;R3=RHDS1
210                                                             ;R4=RHER1
211 010702  005037  001406              CLR     ERFLG$  ;CLEAR ERROR FLAG
212
213                             ;SIMULATED DISK AREA WILL BE USED AS A TEMPORARY
214                             ;STORAGE TABLE FOR DRIVES PRESENT DETERMINED FROM 'NED' = 0 IN RHCS2
215
216 010706  012701  050224              MOV     #DISK,R1        ;LOAD TABLE POINTER
217 010712  012700  000010              MOV     #8.,R0          ;LOAD TABLE LOCATION COUNTER
218 010716  012721  177777      1$:     MOV     #-1,(R1)+       ;FILL 8 LOCATIONS WITH -1
219 010722  005300                      DEC     R0              ;COUNT DOWN ONE LOCATION
220 010724  001374                      BNE     1$              ;BRANCH IF 8 NOT DONE
221
222 010726  005012                      CLR     @R2             ;SELECT UNIT NO.0 (U2!U1!U0=0)
223 010730  012700  000010              MOV     #8.,R0          ;RELOAD TABLE LOCATION COUNTER
```

```
                                                    ;RELOAD THE TABLE POINTER
                                                    ;READ A DRIVE REGISTER (RHER1)
                                                    ;NON EXISTENT DRIVE BIT = 0 ?
                                                    ;YES...DRIVE PRESENT, CHECK THE TYPE
                                                    ;NO...DECREMENT DRIVE COUNT
                                                    ;CHECK RESULTS IF 8 DRIVES DONE

                                                    ;PUT RHCS2 ON THE STACK
                                                    ;MASK ALL BUT THE UNIT NUMBER
                                                    ;INCREMENT THE UNIT NUMBER
                                                    ;GET RHCS1 ADDRESS
                                                    ;ADDRESS UPPER BYTE OF RHCS1
                                                    ;SET 'TRE' IN RHCS1
                                                    ;WITHOUT ADDRESSING DRIVE
                                                    ;RHCS2 HAS THE INCREMENTED UNIT
                                                    ;WITH 'NED' CLEARED
                                                    ;TEST FOR NEXT DRIVE

;CHECK THE UNIT TYPE AND BUILD 'NED' DERIVED UNITS TABLE

                                                    ;IS THIS A DUAL PORT RP04 ?
                                                    ;ENTER IN TABLE IF SO
                                                    ;IS THIS A SINGLE PORT RP04 ?
                                                    ;ENTER IN TABLE IF SO

                                                    ;IS THIS A DUAL PORT RP06 ?
                                                    ;ENTER IN TABLE IF SO
                                                    ;IS THIS A SINGLE PORT RP06 ?
                                                    ;ENTER IN TABLE IF SO

                                                    ;IS THIS A DUAL PORT RP05 ?
                                                    ;ENTER IN TABLE IF SO
                                                    ;IS THIS A SINGLE PORT RP05 ?
                                                    ;ENTER IN TABLE IF SO
                                                    ;NO RP04 FOUND SO CHECK NEXT UNIT

                                                    ;LOAD MAX NO. OF DRIVES
                                                    ;(SP) NOW HAS THE PRESENT DRIVE NO.
                                                    ;LOAD TABLE, INCR TABLE LOCATION &
                                                    ;RESTORE THE STACK TO WHERE IT WAS
                                                    ;DECREMENT THE DRIVE COUNT
                                                    ;CHECK RESULTS IF 8 UNITS CHECKED
                                                    ;SELECT NEXT UNIT
                                                    ;GO TEST IT

;COMPARE 'NED' DERIVED UNITS TABLE WITH THAT DERIVED USING RHAS IN T4

                                                    ;COMPARE RESULTS
                                                    ;RHER1/RHAS DERIVED DATA
                                                    ;'NED' TEST DATA
                                                    ;NO.OF WORDS TO COMPARE
                                                    ;RETURN FOR ERROR HEADER
                                                    ;RETURN FOR ERROR DATA
                                                    ;RETURN FOR GOOD COMPARISON (NEXT TEST)

;SPECIAL 'NED'/'RHAS' TABLE TYPE OUT ROUTINE (BYPASSES .$ERRTYP AND
;HENCE IGNORES INHIBIT ERROR TYPEOUT SWITCH)
```

```
011124                             5$:
011124  104022                             EMT     22
011126  012703  000010                     MOV     #8.,R3            ;LENGTH OF BOTH UNIT TABLES
011132  012701  001354                     MOV     #UNITS,R1         ;ADDRESS OF RHAS/RHER1 UNITS TABLE
011136  012702  050:24                     MOV     #DISK,R2          ;ADDRESS OF 'NED' RHCS2 UNITS TABLE
011142  012137  001124             14$:    MOV     (R1)+,$GDDAT      ;LOAD RHAS UNIT NO.INTO "$GDDAT" AND
                                                                     ;INCREMENT THE TABLE LOCATION
011146  012237  001126                     MOV     (R2)+,$BDDAT      ;LOAD 'NED' UNIT NO. INTO "$BDDAT"
                                                                     ;& INCR TABLE LOCATION

011152  032777  020000  167760     6$:     BIT     #SW13,@SWR        ;INHIBIT ERROR TYPE OUTS ?
011160  001043                             BNE     13$               ;YES...EXIT
011162  022737  177777  001124             CMP     #-1,$GDDAT        ;DOES RHAS UNIT TABLE LOCATION = -1 ?
011170  001413                             BEQ     11$               ;YES...DON'T TYPE IT - CHECK 'NED' TABLE
011172  104401  063174                     TYPE    ,BLNKS8           ;NO...TAB OVER PC COLUMN
011176  104401  063174                     TYPE    ,BLNKS8           ;TAB OVER THE TEST NO. COLUMN
011202  013746  001124                     MOV     $GDDAT,-(SP)      ;;SAVE $GDDAT FOR TYPEOUT
011206  104403                             TYPOS                     ;;GO TYPE--OCTAL ASCII
011210     006                             .BYTE   6                 ;;TYPE 6 DIGITS
011211     000                             .BYTE   0                 ;;SUPPRESS LEADING ZEROS
011212  104401  063202                     TYPE    ,BLNKS2           ;SPACE OVER TO THE NEXT COLUMN
011216  000406                             BR      12$               ;CHECK THE 'NED' UNIT TABLE

011220  104401  063174             11$:    TYPE    ,BLNKS8           ;TAB OVER THE PC COMUMN
011224  104401  063174                     TYPE    ,BLNKS8           ;TAB OVER THE TEST NO. COLUMN
011230  104401  063174                     TYPE    ,BLNKS8           ;TAB OVER THE RHAS UNIT COLUMN

011234  022737  177777  001126     12$:    CMP     #-1,$BDDAT        ;DOES 'NED' UNIT TABLE LOCATION = - 1 ?
011242  001404                             BEQ     9$                ;YES...DON'T TYPE IT
011244  013746  001126                     MOV     $BDDAT,-(SP)      ;;SAVE $BDDAT FOR TYPEOUT
011250  104403                             TYPOS                     ;;GO TYPE--OCTAL ASCII
011252     006                             .BYTE   6                 ;;TYPE 6 DIGITS
011253     000                             .BYTE   0                 ;;SUPPRESS LEADING ZEROS

011254  104401  001223             9$:     TYPE    ,$CRLF            ;FOR THE NEXT LINE IN BOTH TABLES
011260  005303                             DEC     R3                ;COUNT DOWN 2 TABLES LOCATION COUNTER
011262  001327                             BNE     14$               ;IF NOT = 0 TYPE OUT NEXT 2 LOCATIONS
011264  062706  000014                     ADD     #14,SP            ;ADJUST STACK FOR NO "POP" & RTS FROM "COMPAR"

011270                             13$:


                                   ;;*******************************************************************
                                   ;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
                                ;;*****************************************************************
    011270  000004              TST24:  SCOPE
345 011272  004737  041460              JSR     PC,CLDISK       ;SET REGISTERS AND CLEAR
346 011276  005037  001406              CLR     ERFLG$  ;CLEAR ANY ERRORS
347
348                             ;FILL ALL POSSIBLE BITS WITH ONES
349
350 011302  012777  177777  167720          MOV     #177777,@RHDB   ;BUS ADDRESS REGISTER GETS 177777
351 011310  012777  177777  167714          MOV     #177777,@RHWC   ;WORD COUNT REGISTER GETS 177777
352 011316  012777  177777  167710          MOV     #177777,@RHBA   ;BUS ADDRESS REGISTER GETS 177777
353 011324  052777  157010  167704          BIS     #157010,@RHCS2  ;CONTROL AND STATUS 2 GETS 157010
354 011332  012777  001476  167700          MOV     #1476,@RHCS1    ;CONTROL AND STATUS REGISTER/GETS 1476
355 011340  012777  177777  167674          MOV     #177777,@RHER1  ;ERROR REGISTER1 GETS 177777
356 011346  012777  017437  167670          MOV     #17437,@RHDST   ;DESIRED SECTOR TRACK
357 011354  012777  177777  167664          MOV     #177777,@RHER2  ;ERROR REGISTER 2
358 011362  012777  016277  167660          MOV     #16277,@RHOF    ;OFFSET REGISTER
359 011370  012777  177777  167654          MOV     #177777,@RHCA   ;DESIRED CYLINDER
360 011376  012777  177777  167650          MOV     #177777,@RHER3  ;ERROR REGISTER 3
361 011404  012777  000001  167646          MOV     #DMD,@RHMR      ;MAINTENANCE REGISTER
362 011412  012777  177777  167640          MOV     #177777,@RHMR   ;MAINTENANCE REGISTER
363
364 011420  052712  000040              BIS     #CLR,@R2        ;CLEAR ALL POSSIBLE BITS
365 011424  013712  001374              MOV     UNIT,@R2        ;REINSTATE UNIT NO.
366 011430  012700  001230              MOV     #RHDB,R0        ;R0 CONTAINS ADDR. OF ADDR. OF REG.
367
368                             ;DATA BUFFER REGISTER
369
370 011434  012737  177777  001124          MOV     #177777,$GDDAT;GOOD DATA FOR ERROR
371 011442  011037  041130              MOV     @R0,REGADR      ;REGISTER ADDRESS
372 011446  013037  001126              MOV     @(R0)+,$BDDAT   ;TEST DATA
373 011452  023737  001124  001126      CMP     $GDDAT,$BDDAT;COMPARE GOOD WITH TEST DATA
374 011460  001401                      BEQ     2$              ;BRANCH IF GOOD
375 011462  104001                      EMT     1
376                                                             ;AFTER A CLR IN RHCS2
377
378 011464  052712  000040          2$:     BIS     #CLR,@R2        ;SET CLEAR AGAIN BECAUSE
379                                                             ;READING RHDB AFTER CLEARING  WILL
380                                                             ;SET DLT SC AND TRE
381 011470  013712  001374              MOV     UNIT,@R2        ;REINSTATE UNIT NO.
382
383                             ;WORD COUNT REGISTER
384
385
    011474  012737  177777  001124  3$:     MOV     #177777,$GDDAT  ;GOOD DATA FOR ERROR TYPEOUT
```

```
011502  013037  001126                  MOV     @(R0)+,$BDDAT   ;TEST DATA
011506  022737  177777  001126          CMP     #177777,$BDDAT  ;COMPARE DATA
011514  001402                          BEQ     4$              ;BRANCH IF GOOD
011516  004737  012302                  JSR     PC,ERCS2C       ;JUMP TO ERROR FOR CLR (BIT 5)
                                                                ;IN RHCS2
386
387                             ;BUS ADDRESS REGISTER
388
389
011522  012737  000000  001124  4$:     MOV     #0,$GDDAT       ;GOOD DATA FOR ERROR TYPEOUT
011530  013037  001126                  MOV     @(R0)+,$BDDAT   ;TEST DATA
011534  022737  000000  001126          CMP     #0,$BDDAT       ;COMPARE DATA
011542  001402                          BEQ     5$              ;BRANCH IF GOOD
011544  004737  012302                  JSR     PC,ERCS2C       ;JUMP TO ERROR FOR CLR (BIT 5)
                                                                ;IN RHCS2
390
391                             ;CONTROL AND STATUS 2 REGISTER
392
393
011550  012746  000100          5$:     MOV     #100,-(SP)      ;INCLUDE IR
011554  053716  001374                  BIS     UNIT,(SP)       ;SET UNIT NO.
011560  012637  001124                  MOV     (SP)+,$GDDAT    ;GOOD DATA FOR TYPE OUT
011564  013037  001126                  MOV     @(R0)+,$BDDAT   ;TEST DATA
011570  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE DATA
011576  001402                          BEQ     6$              ;BRANCH IF GOOD
011600  004737  012302                  JSR     PC,ERCS2C       ;JUMP TO ERROR FOR CLR (BIT 5)
                                                                ;IN RHCS2
394
395                             ;CONTROL AND STATUS 1 REGISTER
396
397
011604  012737  004276  001124  6$:     MOV     #4276,$GDDAT    ;GOOD DATA FOR ERROR TYPEOUT
011612  013037  001126                  MOV     @(R0)+,$BDDAT   ;TEST DATA
011616  022737  004276  001126          CMP     #4276,$BDDAT    ;COMPARE DATA
011624  001402                          BEQ     7$              ;BRANCH IF GOOD
011626  004737  012302                  JSR     PC,ERCS2C       ;JUMP TO ERROR FOR CLR (BIT 5)
                                                                ;IN RHCS2
398
399                             ;ERROR 1 REGISTER
400
401
011632  012737  000000  001124  7$:     MOV     #0,$GDDAT       ;GOOD DATA FOR ERROR TYPEOUT
011640  013037  001126                  MOV     @(R0)+,$BDDAT   ;TEST DATA
011644  022737  000000  001126          CMP     #0,$BDDAT       ;COMPARE DATA
011652  001402                          BEQ     10$             ;BRANCH IF GOOD
011654  004737  012302                  JSR     PC,ERCS2C       ;JUMP TO ERROR FOR CLR (BIT 5)
                                                                ;IN RHCS2
402
403                             ;DESIRED SECTOR/TRACK REGISTER
404
405
011660  012737  017437  001124  10$:    MOV     #17437,$GDDAT   ;GOOD DATA FOR ERROR TYPEOUT
011666  013037  001126                  MOV     @(R0)+,$BDDAT   ;TEST DATA
011672  022737  017437  001126          CMP     #17437,$BDDAT   ;COMPARE DATA
011700  001402                          BEQ     11$             ;BRANCH IF GOOD
011702  004737  012302                  JSR     PC,ERCS2C       ;JUMP TO ERROR FOR CLR (BIT 5)
                                                                ;IN RHCS2
```

```
                                        ;ERROR 2 REGISTER

011706  012737  000000  001124  11$:    MOV     #0,$GDDAT       ;GOOD DATA FOR ERROR TYPEOUT
011714  013037  001126                   MOV     @(R0)+,$BDDAT   ;TEST DATA
011720  022737  000000  001126           CMP     #0,$BDDAT       ;COMPARE DATA
011726  001402                           BEQ     12$             ;BRANCH IF GOOD
011730  004737  012302                   JSR     PC,ERCS2C       ;JUMP TO ERROR FOR CLR (BIT 5)
                                                                 ;IN RHCS2

                                        ;OFFSET REGISTER

011734  012737  116000  001124  12$:    MOV     #116000,$GDDAT  ;GOOD DATA FOR ERROR TYPEOUT
011742  013037  001126                   MOV     @(R0)+,$BDDAT   ;TEST DATA
011746  022737  116000  001126           CMP     #116000,$BDDAT  ;COMPARE DATA
011754  001402                           BEQ     13$             ;BRANCH IF GOOD
011756  004737  012302                   JSR     PC,ERCS2C       ;JUMP TO ERROR FOR CLR (BIT 5)
                                                                 ;IN RHCS2

                                        ;DESIRED CYLINDER ADDRESS REGISTER

011762  012737  001777  001124  13$:    MOV     #1777,$GDDAT    ;GOOD DATA FOR ERROR TYPEOUT
011770  013037  001126                   MOV     @(R0)+,$BDDAT   ;TEST DATA
011774  022737  001777  001126           CMP     #1777,$BDDAT    ;COMPARE DATA
012002  001402                           BEQ     14$             ;BRANCH IF GOOD
012004  004737  012302                   JSR     PC,ERCS2C       ;JUMP TO ERROR FOR CLR (BIT 5)
                                                                 ;IN RHCS2

                                        ;ERROR 3 REGISTER

012010  012737  000000  001124  14$:    MOV     #0,$GDDAT       ;GOOD DATA FOR ERROR TYPEOUT
012016  013037  001126                   MOV     @(R0)+,$BDDAT   ;TEST DATA
012022  022737  000000  001126           CMP     #0,$BDDAT       ;COMPARE DATA
012030  001402                           BEQ     15$             ;BRANCH IF GOOD
012032  004737  012302                   JSR     PC,ERCS2C       ;JUMP TO ERROR FOR CLR (BIT 5)
                                                                 ;IN RHCS2

                                        ;ATTENTION SUMMARY REGISTER

012036  013037  001126           15$:    MOV     @(R0)+,$BDDAT           ;GET RHAS CONTENTS
012042  012737  000000  001124           MOV     #0,$GDDAT               ;GOOD DATA FOR ERROR TYPE OUT
012050  123737  001124  001126           CMPB    $GDDAT,$BDDAT   ;COMPARE FOR RHAS
012056  001402                           BEQ     16$             ;BRANCH IF GOOD
012060  004737  012302                   JSR     PC,ERCS2C       ;JUMP TO ERROR FOR CLR (BIT 5)
                                                                 ;IN RHCS2

                                        ;MAINTAINABILITY REGISTER

012064  012737  000400  001124  16$:    MOV     #400,$GDDAT     ;GOOD DATA FOR ERROR TYPEOUT
012072  013037  001126                   MOV     @(R0)+,$BDDAT   ;TEST DATA
012076  022737  000400  001126           CMP     #400,$BDDAT     ;COMPARE DATA
012104  001402                           BEQ     17$             ;BRANCH IF GOOD
```

```
012106  004737  012302                  JSR     PC,ERCS2C       ;JUMP TO ERROR FOR CLR (BIT 5)
                                                                ;IN RHCS2
435
436                             ;DRIVE STATUS REGISTER
437
438 012112  012737  000600  001124  17$:   MOV     #600,$GDDAT     ;GOOD DATA FOR ERROR TYPEOUT
439 012120  013046                        MOV     @(R0)+,-(SP)    ;GET RHDS1
440 012122  011637  001126                MOV     (SP),$BDDAT     ;TEST DATA
441 012126  042716  001100                BIC     #VV!PROG,(SP)   ;CLEAR VV AND PROG
442 012132  022726  000600                CMP     #600,(SP)+      ;COMPARE DATA
443 012136  001402                        BEQ     20$             ;BRANCH IF GOOD
444 012140  004737  012302                JSR     PC,ERCS2C       ;JUMP TO ERROR FOR CLR (BIT 5)
445                                                             ;IN RHCS2
446
447                             ;DRIVE TYPE
448
449
012144  013737  001410  001124  20$:   MOV     SAVDT,$GDDAT    ;GOOD DATA FOR ERROR TYPEOUT
012152  013037  001126                MOV     @(R0)+,$BDDAT   ;TEST DATA
012156  023737  001410  001126        CMP     SAVDT,$BDDAT    ;COMPARE DATA
012164  001402                        BEQ     21$             ;BRANCH IF GOOD
012166  004737  012302                JSR     PC,ERCS2C       ;JUMP TO ERROR FOR CLR (BIT 5)
                                                                ;IN RHCS2
450
451                             ;SERIAL NUMBER REGISTER
452
453
454
012172  013737  001412  001124  21$:   MOV     SAVSN,$GDDAT    ;GOOD DATA FOR ERROR TYPEOUT
012200  013037  001126                MOV     @(R0)+,$BDDAT   ;TEST DATA
012204  023737  001412  001126        CMP     SAVSN,$BDDAT    ;COMPARE DATA
012212  001402                        BEQ     22$             ;BRANCH IF GOOD
012214  004737  012302                JSR     PC,ERCS2C       ;JUMP TO ERROR FOR CLR (BIT 5)
                                                                ;IN RHCS2
455
456                             ;ECC1 POSITION
457
458
012220  012737  000000  001124  22$:   MOV     #0,$GDDAT       ;GOOD DATA FOR ERROR TYPEOUT
012226  013037  001126                MOV     @(R0)+,$BDDAT   ;TEST DATA
012232  022737  000000  001126        CMP     #0,$BDDAT       ;COMPARE DATA
012240  001402                        BEQ     23$             ;BRANCH IF GOOD
012242  004737  012302                JSR     PC,ERCS2C       ;JUMP TO ERROR FOR CLR (BIT 5)
                                                                ;IN RHCS2
459
460
461                             ;ECC2 PATTERN
462
463
012246  012737  000000  001124  23$:   MOV     #0,$GDDAT       ;GOOD DATA FOR ERROR TYPEOUT
012254  013037  001126                MOV     @(R0)+,$BDDAT   ;TEST DATA
012260  022737  000000  001126        CMP     #0,$BDDAT       ;COMPARE DATA
012266  001402                        BEQ     24$             ;BRANCH IF GOOD
012270  004737  012302                JSR     PC,ERCS2C       ;JUMP TO ERROR FOR CLR (BIT 5)
                                                                ;IN RHCS2
464
465
```

```
                                        ;LOOK-AHEAD REGISTER
                                        ;
012274  005720                  24$:    TST     (R0)+           ;AS THE LOOK-AHEAD REG. CANNOT BE PREDICTED
                                        ;AFTER AN INIT IT IS NOT CHECKED
                                        ;
                                        ;CURRENT CYLINDER ADDRESS REGISTER
                                        ;
012276  005720                  25$:    TST     (R0)+           ;AS THE CURRENT REG. CANNOT BE PREDICTED
                                        ;AFTER A INIT IT IS NOT CHECKED
                                        ;
012300                          26$:
012300  000405                          BR      TST25           ;BRANCH OVER JSR
                                        ;
                                        ;
012302  014037  041130          ERCS2C: MOV     -(R0),REGADR    ;FAILING REGISTER ADDRESS
012306  104001                          EMT     1
                                                                ;NOT CLEAR APPROPIATE BITS
                                                                ;OR CLEARED EXTRA BITS
012310  005720                          TST     (R0)+           ;UNDO -(R0) FOR BAD DATA
012312  000207                          RTS     PC              ;RETURN TO TEST ABOVE
                                        ;
                                        ;;*****************************************************************
                                        ;*TEST 25       PACK ACKNOWLEDGE COMMAND TEST
                                        ;*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
                                        ;;*****************************************************************
012314  000004                  TST25:  SCOPE
012316  012706  001100                  MOV     #STACK,SP       ;RESET STACK
012322  012737  000025  001432          MOV     #25,TSTNM       ;MOVE #25 TO TEST NUMBER
                                        ;
012330  004737  041460                  JSR     PC,CLDISK       ;INIT AND SET UP GENERAL CPU/DEVICE
                                                                ;REGISTER CORRESPONDENCE AND UNIT NO.
012334  012777  000001  166716          MOV     #DMD,@RHMR      ;SET DIAGNOSTIC MODE
012342  013777  001500  166670          MOV     PKACK,@RHCS1    ;LOAD "PACK ACKNOWLEDGE COMMAND" INTO RHCS1
                                        ;
                                        ;SAVE REGISTERS FOR COMPARISON AFTER 'GO' IS ISSUED
012350  004037  042152                  JSR     R0,SAVER        ;SAVE
012354  001232                          RHWC                    ;FROM
012356  002554                          REINTO                  ;TO
012360  000023                          19.                     ;NUMBER OF REGISTERS SAVED
                                        ;
                                        ;
012362  052777  000001  166650          BIS     #GO,@RHCS1      ;ISSUE 'GO' TO PACK ACKNOWLEDGE COMMAND
                                        ;
                                        ;CHANGE SAVED REGISTERS TO EXPECTED VALUES
012370  052737  000100  002604          BIS     #VV,REINTO+30   ;SAVED RHDS1
                                        ;
                                        ;AFTER GO HAS BEEN GIVEN TO PACK ACKNOWLEDGE COMMAND
                                        ;SAVE REGISTERS AGAIN SO THAT COMPARISONS CAN
                                        ;BE DONE
                                        ;
012376  004037  042152                  JSR     R0,SAVER        ;SAVE
012402  001232                          RHWC                    ;FROM
012404  001510                          WRFROM
012406  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
012410  113737  002601  001535                MOVB    REINTO+25,WRFROM+25;SAVE UPPER RHAS

                                        ;COMPARE REGISTERS BEFORE PACK ACKNOWLEDGE COMMAND
                                        ;WITH AFTER GO

012416  004037  042354                        JSR     R0,COMPAR       ;COMPARE
012422  002554                                REINTO                  ;GOOD BUFFER
012424  001510                                WRFROM                  ;TEST BUFFER
012426  000023                                19.                     ;NUMBER
012430  012436                                1$                      ;RETURN FOR ERROR
012432  012436                                1$                      ;SAME
012434  012456                                2$                      ;RETURN FOR GOOD COMPARISON

012436  013705  046426          1$:           MOV     ERWORD,R5       ;GETTING READY TO INDEX
012442  060505                                ADD     R5,R5           ;DOUBLE ERROR WORD
012444  016537  001230  041130                MOV     RHWC-2(R5),REGADR ;FAILING REGISTER ADDRESS

012452  104001                                EMT     1
                                                                      ;AFTER PACK ACKNOWLEDGE COMMAND
                                                                      ;WITH GO IS GIVEN
012454  000207                                RTS     PC              ;RETURN TO COMPARISON

012456                                  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
                                        ;;**********************************************************************
012456  000004                        TST26:  SCOPE
012460  012706  001100                        MOV     #STACK,SP       ;RESET STACK
012464  012737  000026  001432                MOV     #26,TSTNM       ;MOVE #26 TO TEST NUMBER
012472  004737  041460                        JSR     PC,CLDISK       ;INIT AND SET UP GENERAL CPU/DEVICE
                                                                      ;REGISTER CORRESPONDENCE

                                        ;FILL ALL POSSIBLE REGISTER BITS WITH ONES

012476  012777  177777  166526                MOV     #177777,@RHWC   ;WORD COUNT REGISTER GETS 177777
012504  012777  177777  166522                MOV     #177777,@RHBA   ;BUS ADDRESS REGISTER GETS 177777
012512  052777  157010  166516                BIS     #157010,@RHCS2  ;CONTROL AND STATUS 2 GETS 177430
012520  012777  001476  166512                MOV     #1476,@RHCS1    ;CONTROL AND STATUS REGISTER 1 GETS 21476
012526  012777  177777  166506                MOV     #177777,@RHER1  ;ERROR REGISTER1 GETS 177777
012534  012777  017437  166502                MOV     #17437,@RHDST   ;DESIRED SECTOR TRACK
012542  012777  177777  166476                MOV     #177777,@RHER2  ;ERROR REGISTER 2
012550  012777  016277  166472                MOV     #16277,@RHOF    ;OFFSET REGISTER
012556  012777  000777  166466                MOV     #777,@RHCA      ;DESIRED CYLINDER
012564  012777  177777  166462                MOV     #177777,@RHER3  ;ERROR REGISTER 3
012572  012777  000001  166460                MOV     #DMD,@RHMR      ;MAINTENANCE REGISTER
012600  012777  177777  166452                MOV     #177777,@RHMR   ;MAINTENANCE REGISTER

                                        ;BEFORE RESET SAVE REGISTERS IN READ INTO BUFFER
```

```
577 012606  004037  042152                  JSR     R0,SAVER        ;SAVE
578 012612  001232                          RHWC                    ;FROM
579 012614  002554                          REINTO                  ;TO
580 012616  000021                          17.                     ;NUMBER
581
582                                 ;GIVE RESET AND REINSTATE UNIT NUMBER
583 012620  000005                          RESET
584 012622  004737  054216                  JSR     PC,$TKINT       ;INITIALIZE TK
585 012626  053777  001374  166402          BIS     UNIT,@RHCS2
586
587                                 ;CHANGE ORIGINAL SAVED REGISTERS TO EXPECTED VALUES AFTER RESET
588 012634  005037  002556                  CLR     REINTO+2        ;CLEAR SAVED RHBA
589 012640  013746  001374                  MOV     UNIT,-(SP)      ;GET UNIT NUMBER FRO SAVED RHCS2
590 012644  052716  000100                  BIS     #IR,(SP)        ;INCLUDE IR
591 012650  012637  002560                  MOV     (SP)+,REINTO+4 ;SAVED RHCS2
592 012654  012737  004276  002562          MOV     #DVA!RDY!76,REINTO+6 ;SAVED RHCS1
593 012662  005037  002564                  CLR     REINTO+10       ;SAVED RHER1
594 012666  005037  002570                  CLR     REINTO+14       ;SAVED RHER2
595 012672  012737  116000  002572          MOV     #116000,REINTO+16 ;SAVED RHOF
596 012700  005037  002576                  CLR     REINTO+22       ;SAVED RHER3
597 012704  105037  002600                  CLRB    REINTO+24       ;SAVED RHAS
598 012710  012737  000400  002602          MOV     #400,REINTO+26;SAVED RHMR
599
600                                 ;CHANGE RHDS1 WITHOUT CHANGING PROG BIT
601 012716  013746  002604                  MOV     REINTO+30,-(SP) ;GET RHDS1
602 012722  042716  176777                  BIC     #^CPROG,(SP)    ;CLEAR EVERYTHING EXCEPT PROG
603 012726  052716  000700                  BIS     #700,(SP)       ;SET EXPECTED BITS - 'DPR', 'DRY' & 'VV'
604
605 012732  012637  002604          4$:     MOV     (SP)+,REINTO+30;SAVED RHDS1
606 012736  005037  002612                  CLR     REINTO+36       ;SAVED RHEC1
607 012742  005037  002614                  CLR     REINTO+40       ;SAVED RHEC2
608
609                                 ;AFTER RESET, SAVE REGISTERS FOR COMPARISONS TO BE DONE
610 012746  004037  042152                  JSR     R0,SAVER        ;SAVE
611 012752  001232                          RHWC                    ;FROM
612 012754  001510                          WRFROM                  ;TO
613 012756  000021                          17.                     ;NUMBER
614
615                                 ;AS UPPER BYTE OF RHAS CAN BE CHANGING IN A DUAL PORT
616                                 ;OPERATION THE UPPER BYTE OF RHAS WILL BE SAVED AS IS
617                                 ;SO THAT THE COMPARES ARE ONLY VALID FOR THE LOWER BYTE
618 012760  113737  002601  001535          MOVB    REINTO+25,WRFROM+25;SAVE UPPER RHAS
619
620                                 ;COMPARE REGISTERS BEFORE RESET WITH REGISTERS AFTER RESET
621 012766  004037  042354                  JSR     R0,COMPAR       ;COMPARE
622 012772  002554                          REINTO                  ;GOOD BUFFER
623 012774  001510                          WRFROM                  ;TEST BUFFER
624 012776  000021                          17.                     ;NUMBER
625 013000  013006                          1$                      ;RETURN FOR ERROR
626 013002  013006                          1$                      ;SAME
627 013004  013026                          2$                      ;RETURN FOR GOOD COMPARISON
628
629 013006  013705  046426          1$:     MOV     ERWORD,R5       ;GETTING READY TO INDEX
630 013012  060505                          ADD     R5,R5           ;DOUBLE ERROR WORD
631 013014  016537  001230  041130          MOV     RHWC-2(R5),REGADR ;FAILING REGISTER ADDRESS
632 013022  104001                          EMT     1
633                                                                 ;A RESET THAT IS AN
```

```
                                                        ;UNIBUS INITIALIZE CAUSED
                                                        ;AN IMPROPER REGISTER CHANGE
013024  000207                      RTS     PC          ;RETURN TO COMPARISON
013026                          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
                                ;;*********************************************************************
013026  000004                  TST27:  SCOPE
013030  012706  001100                  MOV     #STACK,SP       ;RESET STACK
013034  012737  000027  001432          MOV     #27,TSTNM       ;MOVE #27 TO TEST NUMBER

                                ;CHECK TO SEE IF THE PROGRAM IS RUNNING WITH AN RH70

013042  005737  001436                  TST     RH70            ;TEST FLAG FOR RH70 CONTROLLER
013046  001402                          BEQ     64$             ;IF FLAG = 1, THIS TEST IS SKIPPED
013050  000137  013204                  JMP     TST30           ;JUMP TO NEXT TEST
013054                          64$:                            ;IF FLAG = 0, DO THIS TEST
013054  004737  041460                  JSR     PC,CLDISK       ;CLEAR DISK AND LOAD R'S

013060  017700  166144                  MOV     @RHDB,R0        ;READ FROM EMPTY SILO
013064  013746  001374                  MOV     UNIT,-(SP)      ;GET UNIT NO. IN
013070  052716  100100                  BIS     #DLT!IR,(SP)    ;GET DATA LATE BIT AND IR
013074  004737  041064                  JSR     PC,PUTREG       ;SAVE REGISTERS
013100  022637  001312                  CMP     (SP)+,CS2       ;IS DATA LATE BIT UP?
013104  001403                          BEQ     1$              ;IF YES BRANCH
013106  010237  001122                  MOV     R2,$BDADR       ;IF NOT STORE FAILING REG.
013112  104011                          EMT     11
                                                                ;RHCS2 SHOULD HAVE ONLY
                                                                ;DLT AND UNIT NUMBER (BIT 0-2)
                                                                ;ALL OTHER BITS SHOULD
                                                                ;BE 0
013114  022737  144200  001314  1$:     CMP     #SC!TRE!RDY!DVA,CS1 ;IS SPECIAL CONDITION, TRANSFER ERROR
                                                                ;AND READY UP?
013122  001403                          BEQ     2$              ;IF YES BRANCH
013124  010137  001122                  MOV     R1,$BDADR       ;IF NOT STORE FAILING REG.
013130  104011                          EMT     11
                                                                ;TRE AND RDY. AFTER A
                                                                ;READ FROM EMPTY SILO ONLY
                                                                ;THESE BITS SHOULD BE UP
                                                                ;ALL OTHERS SHOULD BE 0
013132  012711  040000          2$:     MOV     #TRE,@R1        ;CLEAR ERROR BITS BY MOVING
                                                                ;ONE INTO TRE IN RHCS1
013136  004737  041064                  JSR     PC,PUTREG       ;SAVE REGISTERS
013142  022737  004200  001314          CMP     #RDY!DVA,CS1    ;ALL BITS BUT RDY AND DVA SHOULD
                                                                ;BE 0
013150  001403                          BEQ     3$              ;BRANCH IF YES
013152  010137  001122                  MOV     R1,$BDADR       ;STORE FAILING ADDRESS
013156  104011                          EMT     11
                                                                ;READY AND DVA SHOULD BE SET IN
                                                                ;RHCS1
013160  013746  001374          3$:     MOV     UNIT,-(SP)
013164  052716  000100                  BIS     #IR,(SP)
013170  022637  001312                  CMP     (SP)+,CS2       ;RHCS2 SHOULD HAVE IR AND UNIT ONLY
013174  001403                          BEQ     TST30           ;BRANCH IF YES
```

```
                                ;;*****************************************************************
                                ;*TEST 30        SILO TEST 2
                                ;*THIS TESTS THE IR AND "OR" BITS OF RHCS2
                                ;*AT THE BEGINNING IR SHOULD BE SET AND "OR" RESET
                                ;*LOADING 0 IN SILO RESETS IR FOR ONLY 2 MICRO SECONDS
                                ;*THIS TIME CANNOT BE CHECKED BUT IT IS CHECKED TO SEE IF
                                ;*IT DOES GO DOWN OR NOT
                                ;*THEN ALL 1 IS LOADED IN SILO "OR" SHOULD BECOME SET
                                ;*IN 30 MICRO SECONDS AGAIN TIME IS NOT  CHECKED
                                ;*"OR" SHOULD BE SET
                                ;*THE OUTPUT FROM THE SILO SHOULD BE 0 AND ALL ONES
                                ;;*****************************************************************
013204  000004                TST30:  SCOPE
 66 013206  012706  001100              MOV     #STACK,SP       ;RESET STACK
 67 013212  012737  000030  001432      MOV     #30,TSTNM       ;MOVE #30 TO TEST NUMBER
 68
                                ;CHECK TO SEE IF THE PROGRAM IS RUNNING WITH AN RH70

013220  005737  001436                  TST     RH70            ;TEST FLAG FOR RH70 CONTROLLER
013224  001402                          BEQ     64$             ;IF FLAG = 1, THIS TEST IS SKIPPED
013226  000137  013432                  JMP     TST31           ;JUMP TO NEXT TEST
013232                          64$:                            ;IF FLAG = 0, DO THIS TEST
 69 013232  004737  041460              JSR     PC,CLDISK       ;CLEAR REGISTERS LOAD R'S
 70
 71 013236  013746  001374              MOV     UNIT,-(SP)
 72 013242  052716  000100              BIS     #IR,(SP)
 73 013246  004737  041064              JSR     PC,PUTREG       ;SAVE REGISTERS
 74 013252  022637  001312              CMP     (SP)+,CS2       ;IR SHOULD BE SET "OR" RESET
 75 013256  001403                      BEQ     1$
 76 013260  010237  001122              MOV     R2,$BDADR       ;FAILING REGISTER RHCS2
 77 013264  104011                      EMT     11
 78                                                             ;SET, UNIT NO. SET AND
 79                                                             ;ALL OTHER BITS 0
 80 013266  005077  165736          1$:     CLR     @RHDB           ;LOAD DATA BUFFER (SILO) WITH 0
 81 013272  012777  177777  165730      MOV     #-1,@RHDB       ;LOAD SILO WITH ALL ONES
 82 013300  013737  001236  013310      MOV     RHCS2,2$        ;ADDRESS OF RHCS2
 83 013306  104415                      WAT                     ;WAIT TRAP
 84 013310  000000                  2$:     .WORD               ;ADDRESS OF RHCS2
 85 013312  000200                      OR
 86 013314  013746  001374          3$:     MOV     UNIT,-(SP)      ;
 87 013320  052716  000300              BIS     #OR!IR,(SP)     ;IR AND "OR"
 88 013324  004737  041064              JSR     PC,PUTREG       ;SAVE REGISTERS
 89 013330  022637  001312              CMP     (SP)+,CS2       ;IR AND "OR" SHOULD BE SET
 90 013334  001403                      BEQ     4$
 91 013336  010237  001122              MOV     R2,$BDADR       ;SAVE RHCS2 ADDR. FAILING REG.
 92 013342  104011                      EMT     11
 93                                                             ;SET TOGETHER WITH IR AND
 94                                                             ;UNIT NO.
 95 013344  017700  165660          4$:     MOV     @RHDB,R0        ;SAVE SILO DATA SHOULD BE 0
 96 013350  017705  165654              MOV     @RHDB,R5        ;SAVE SILO DATA SHOULD BE ALL 1
 97 013354  022700  000000              CMP     #0,R0           ;FIRST WORD 0? XYZ DO MORE TEST
 98 013360  001410                      BEQ     5$              ;BRANCH IF YES
 99 013362  005037  001124              CLR     $GDDAT  ;GOOD DATA
100
```

```
                                                MOV     R0,$BDDAT       ;BAD DATA
                                                MOV     RHDB,REGADR     ;SAVE RHDB FAILING REG.
                                                EMT     1
                                                                        ;"0" WHEN "DR" WAS SET
                                        5$:     CMP     #-1,R5          ;SECOND WORD ALL ONES?
                                                BEQ     TST31           ;BRANCH IF YES
                                                MOV     #-1,$GDDAT      ;GOOD DATA
                                                MOV     R5,$BDDAT       ;BAD DATA
                                                MOV     RHDB,REGADR     ;SAVE RHDB FAILING REG.
                                                EMT     1
                                                                        ;WORD OF ALL ONES WHEN "DR"
                                                                        ;WAS SET


                                        ;;**********************************************************************
                                        ;*TEST 31       SILO TEST 3
                                        ;*THIS TESTS SILO BUFFER BY FILLING IT WITH A COUNT FROM
                                        ;*0 TO 65 AND THEN CHECKING IF IR IS DOWN AND "DR"
                                        ;*IS HIGH AND COMPARING THE SILO OUTPUT.
                                        ;;**********************************************************************
013432  000004                          TST31:  SCOPE
013434  012737  000031  001432                  MOV     #31,TSTNM       ;MOVE #31 TO TEST NUMBER

                                        ;CHECK TO SEE IF THE PROGRAM IS RUNNING WITH AN RH70

013442  005737  001436                          TST     RH70            ;TEST FLAG FOR RH70 CONTROLLER
013446  001402                                  BEQ     64$             ;IF FLAG = 1, THIS TEST IS SKIPPED
013450  000137  013672                          JMP     TST32           ;JUMP TO NEXT TEST
013454                                  64$:                            ;IF FLAG = 0, DO THIS TEST

013454  012700  050224                          MOV     #SILOTB,R0      ;TABLE POINTER
013460  012705  000103                          MOV     #67.,R5         ;COUNTER
                                        1$:     CLR     (R0)+           ;CLEAR TOTAL TABLE
                                                DEC     R5              ;COUNT
                                                BNE     1$              ;BRANCH IF NOT COMPLETELY CLEAR
                                                JSR     PC,CLDISK       ;CLEAR ALL REG.
                                                CLR     R0
                                                MOV     #66.,R5         ;COUNT
                                        2$:     MOV     R0,@RHDB        ;LOAD SILO WITH COUNT FROM 0 TO 65
                                                INC     R0              ;NEXT COUNT
                                                DEC     R5              ;IS 66 LOADS DONE?
                                                BNE     2$              ;BRANCH IF NOT.
                                                MOV     UNIT,-(SP)
                                                BIS     #OR,(SP)
                                                JSR     PC,PUTREG       ;SAVE REGISTERS
                                                CMP     (SP)+,CS2       ;"DR" SHOULD BE SET IR RESET
                                                BEQ     3$              ;BRANCH IF YES
                                                MOV     R2,$BDADR       ;SAVE RHCS2 ADR. FAILING REG.
                                                EMT     11
                                                CLR     ERFLG$  ;RESET AFTER SILO WAS FULL
                                        3$:     MOV     #SILOTB,R0      ;POINTER
                                                MOV     #66.,R5         ;COUNTER
                                        4$:     MOV     @RHDB,(R0)+     ;READ SILO
                                                DEC     R5              ;COUNT
                                                BNE     4$              ;BRANCH IF 66 NOT DONE
                                                MOV     #SILOTB,R0      ;POINTER
                                                MOV     #66.,R5
                                                CLR     -(SP)
```

```
                                5$:     CMP     (SP),(R0)+
                                        BEQ     8$              ;BRANCH IF GOOD
                                        MOV     -(R0),$BDDAT    ;BAD DATA
                                        MOV     (SP),$GDDAT     ;GOOD DATA
                                        MOV     RHDB,REGADR     ;FAILING REG. RHDB
                                        TST     ERFLG$  ;IS THIS FIRST ERROR?
                                        BNE     6$              ;IF NOT BRANCH
                                        EMT     12
                                        BR      7$              ;BRANCH TO AVOID PRINTING NEXT ERROR
                                6$:
                                        EMT     13
                                                                ;HAD A COUNT WRITTEN IN.
                                                                ;ON READ OUT AN ERROR WAS
                                                                ;DETECTED. THE TOTAL SILO
                                                                ;READOUT IS IN LOCATION
                                                                ;"SILOTB" TO THE NEXT 65
                                                                ;WORDS.
                                7$:     TST     (R0)+           ;INCREMENT (R0)
                                                                ;ARE FURTHER COMPARES TO
                                        MOV     @SWR,-(SP)      ;BE DONE
                                        BIC     #^CSW07!SW08,(SP) ;ONLY KEEP SW7 AND SW8
                                        CMP     #SW07,(SP)+     ;TEST SW07
                                        BEQ     9$              ;IF NO MORE COMPARE THEN BRANCH
                                8$:     INC     (SP)            ;NEXT GOOD WORD
                                        DEC     R5              ;COUNT
                                        BNE     5$              ;BRANCH IF 66 NOT COMPLETE
                                9$:     TST     (SP)+           ;POP STACK

                                ;;*****************************************************************
                                ;*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
                                ;;*****************************************************************
    013672  000004              TST32:  SCOPE
    013674  012737  000032  001432      MOV     #32,TSTNM       ;MOVE #32 TO TEST NUMBER

                                ;CHECK TO SEE IF THE PROGRAM IS RUNNING WITH AN RH70

    013702  005737  001436              TST     RH70            ;TEST FLAG FOR RH70 CONTROLLER
    013706  001402                      BEQ     64$             ;IF FLAG = 1, THIS TEST IS SKIPPED
    013710  000137  014016              JMP     TST33           ;JUMP TO NEXT TEST
    013714                      64$:                            ;IF FLAG = 0, DO THIS TEST
    013714  004737  041460              JSR     PC,CLDISK       ;CLEAR DISK REG.

    013720  005000                      CLR     R0              ;CLEAR R0
    013722  005200                      INC     R0              ;ADD 1
    013724  010077  165300              MOV     R0,@RHDB        ;LOAD SILO
    013730  022700  000103              CMP     #67.,R0         ;67 DONE?
    013734  001401                      BEQ     2$              ;BRANCH IF YES
    013736  000771                      BR      1$              ;NO SO BRANCH
    013740  004737  041064      2$:     JSR     PC,PUTREG       ;SAVE REGISTERS

    013744  032737  100000  001312      BIT     #DLT,CS2        ;DLT SET?
    013752  001003                      BNE     3$              ;BRANCH IF YES
    013754  010237  001122              MOV     R2,$BDADR       ;FAILING ADDRESS RHCS2
    013760  104011                      EMT     11
    013762  017737  165242  001126  3$: MOV     @RHDB,$BDDAT    ;INPUT TO SILO
```

```
                                ;;*****************************************************************
                                ;*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.
                                ;;*****************************************************************
014016  000004                  TST33:  SCOPE
214 014020  012737  000033  001432          MOV     #33,TSTNM       ;MOVE #33 TO TEST NUMBER
215
                                ;CHECK TO SEE IF THE PROGRAM IS RUNNING WITH AN RH70

014026  005737  001436                      TST     RH70            ;TEST FLAG FOR RH70 CONTROLLER
014032  001402                            BEQ     64$             ;IF FLAG = 1, THIS TEST IS SKIPPED
014034  000137  014316                    JMP     TST34           ;JUMP TO NEXT TEST
014040                          64$:                                ;IF FLAG = 0, DO THIS TEST
216 014040  004737  041460                  JSR     PC,CLDISK       ;CLEAR DISK
217
218 014044  013746  001374                  MOV     UNIT,-(SP)      ;GET UNIT NO.
219 014050  052716  000100                  BIS     #IR,(SP)        ;SET INPUT READY
220 014054  004737  041064                  JSR     PC,PUTREG       ;SAVE REGISTERS
221 014060  022637  001312                  CMP     (SP)+,CS2       ;IR SHOULD BE SET "OR" CLEARED
222 014064  001403                          BEQ     1$              ;BRANCH IF GOOD
223 014066  010237  001122                  MOV     R2,$BDADR       ;FAILING REGISTER RHCS2
224 014072  104011                          EMT     11
225                                                                 ;AND ALL OTHER BITS 0
226 014074  013700  001230          1$:     MOV     RHDB,R0 ;R0 HAS RHDB ADDRESS
227 014100  005001                          CLR     R1              ;DATA
228 014102  010110          2$:             MOV     R1,@R0          ;0, THEN 1 THEN 2 THEN 3
229                                                                 ;IN RHDB
230 014104  005201                          INC     R1              ;INCREMENT DATA
231 014106  022701  000004                  CMP     #4,R1           ;IS 4 DONE
232 014112  103373                          BHIS    2$              ;BRANCH IF NOT
233 014114  013737  001236  014124          MOV     RHCS2,3$        ;
234 014122  104415                          WAT                     ;WAIT FOR "OR"
235 014124  000000          3$:             .WORD   0               ;RHCS2 ADDRESS
236 014126  000200                          OR                      ;WAIT ON OR.
237 014130  004737  041460                  JSR     PC,CLDISK       ;CLR IN RHCS2
238 014134  013746  001374                  MOV     UNIT,-(SP)      ;UNIT NO.
239 014140  052716  000100                  BIS     #IR,(SP)
240 014144  004737  041064                  JSR     PC,PUTREG       ;SAVE REGISTERS
241 014150  022637  001312                  CMP     (SP)+,CS2       ;IR SHOULD BE SET "0"=0
242 014154  001403                          BEQ     4$              ;BRANCH IF GOOD
243 014156  010237  001122                  MOV     R2,$BDADR       ;FAILING REGISTER RHCS2
244 014162  104011                          EMT     11
245                                                                 ;AND ALL OTHER BITS 0
246 014164  013700  001230          4$:     MOV     RHDB,R0 ;R0 HAS RHDB ADDRESS
247 014170  012710  000004                  MOV     #4,@R0          ;LOAD 4 IN SILO
248 014174  011201                          MOV     @R2,R1          ;SAVE RHCS2
249 014176  011005                          MOV     @R0,R5          ;READ THE 4 IN SILO
```

```
    250 014200  011003                          MOV     @R0,R3          ;READ SILO TO GET DLT
    251 014202  011204                          MOV     @R2,R4          ;SAVE RHCS2
    252 014204  032701  000200                  BIT     #OR,R1          ;TEST FOR OR IN RHCS2
    253 014210  001424                          BEQ     6$              ;IF OR IS NOT SET BRANCH
    254 014212  022705  000004                  CMP     #4,R5           ;SILO 4 IS NOW COMPARED
    255 014216  001410                          BEQ     5$
    256 014220  010037  041130                  MOV     R0,REGADR       ;SILO ADDRESS
    257 014224  012737  000004  001124          MOV     #4,$GDDAT       ;GOOD DATA
    258 014232  010537  001126                  MOV     R5,$BDDAT       ;BAD DATA
    259 014236  104001                          EMT     1
    260                                                                 ;PUT IN AFTER ''OR'' WAS UP
    261 014240  005703                  5$:     TST     R3              ;IS IT ZERO BECAUSE SILO
    262                                                                 ;IS DESTRUCTIVE READ
    263 014242  001407                          BEQ     6$              ;BRANCH IF GOOD
    264 014244  010037  041130                  MOV     R0,REGADR       ;SILO ADDRESS
    265 014250  005037  001124                  CLR     $GDDAT  ;GOOD DATA
    266 014254  010337  001126                  MOV     R3,$BDDAT       ;BAD DATA
    267 014260  104001                          EMT     1
    268                                                                 ;AFTER THE ONE WORD PUT IN
    269                                                                 ;HAS BEEN TAKEN OUT AS
    270                                                                 ;SILO IS A DESTRUCTIVE READ
    271 014262  032704  100000          6$:     BIT     #DLT,R4         ;
    272 014266  001013                          BNE     TST34           ;BRANCH IF DLT SET
    273 014270  013746  001374                  MOV     UNIT,-(SP)      ;GET UNIT NO
    274 014274  052716  100300                  BIS     #DLT!OR!IR,(SP) ;
    275 014300  012637  001124                  MOV     (SP)+,$GDDAT    ;GOOD DATA
    276 014304  010437  001126                  MOV     R4,$BDDAT       ;BAD DATA
    277 014310  010237  041130                  MOV     R2,REGADR       ;RHCS2 ADDRESS
    278 014314  104001                          EMT     1
```

```
                                .SBTTL  MORE REGISTER TESTS
                                ;
                                ;;*****************************************************************
                                ;*TEST 34       TEST ODD BYTE INSTRUCTION ON RHCS1 - RH11
                                ;*RDY (BIT 07) AND DVA (BIT 11) SHOULD ALWAYS BE SET
                                ;;*****************************************************************
  014316  000004                TST34:  SCOPE
  014320  012737  000034  001432        MOV     #34,TSTNM       ;MOVE #34 TO TEST NUMBER
                                ;
                                ;CHECK TO SEE IF THE PROGRAM IS RUNNING WITH AN RH70
  014326  005737  001436                TST     RH70            ;TEST FLAG FOR RH70 CONTROLLER
  014332  001402                        BEQ     64$             ;IF FLAG = 1, THIS TEST IS SKIPPED
  014334  000137  014454                JMP     TST35           ;JUMP TO NEXT TEST
  014340                        64$:                            ;IF FLAG = 0, DO THIS TEST
  014340  012706  001100                MOV     #STACK,SP       ;RESET STACK
  014344  004737  041460                JSR     PC,CLDISK       ;CLEAR DISK REG.
                                ;
  014350  012711  003566                MOV     #3566,@R1       ;LOAD RHCS1 WITH ANY NUMBER
  014354  010146                        MOV     R1,-(SP)        ;GETTING READY TO FORM ODD BYTE
  014356  005216                        INC     (SP)            ;SP NOW HAS ODD BYTE FOR RHCS1
  014360  112736  000005                MOVB    #5,@(SP)+       ;MOVE 5 INTO ODD BYTE FOR RHCS1
  014364  011137  001126                MOV     @R1,$BDDAT      ;TEST DATA
  014370  022737  006766  001126        CMP     #2566!DVA!RDY,$BDDAT ;RHCS1 SHOULD HAVE 6766
  014376  001406                        BEQ     1$              ;BRANCH IF GOOD
  014400  012737  006766  001124        MOV     #2566!DVA!RDY,$GDDAT ;GOOD DATA
  014406  010137  041130                MOV     R1,REGADR       ;FAILING REGISTER RHCS1
  014412  104001                        EMT     1
                                                                ;ODD BYTE OF RHCS1 GAVE
                                                                ;WRONG RESULTS
                                ;
  014414  112711  000032        1$:     MOVB    #32,@R1         ;MOVE INTO EVEN BYTE
  014420  011137  001126                MOV     @R1,$BDDAT      ;TEST DATA
  014424  022737  006632  001126        CMP     #2432!DVA!RDY,$BDDAT ;RHCS1 SHOULD HAVE 6632
  014432  001460                        BEQ     TST36           ;DO NEXT RH11 TEST IF GOOD
  014434  012737  006632  001124        MOV     #2432!DVA!RDY,$GDDAT ;GOOD DATA
  014442  010137  041130                MOV     R1,REGADR       ;FAILING REGISTER RHCS1
  014446  104001                        EMT     1
                                                                ;BYTE OF RHCS1 GAVE WRONG
                                                                ;RESULT
                                ;
  014450  000137  014574                JMP     TST36           ;SKIP RH70 TEST
                                ;
                                ;;*****************************************************************
                                ;*TEST 35       TEST ODD BYTE INSTRUCTION ON RHCS1 - RH70
                                ;*RDY (BIT 07) AND DVA (BIT 11) SHOULD ALWAYS BE SET
                                ;;*****************************************************************
  014454  000004                TST35:  SCOPE
  014456  012737  000035  001432        MOV     #35,TSTNM       ;MOVE #35 TO TEST NUMBER
  014464  012706  001100                MOV     #STACK,SP       ;RESET STACK
  014470  004737  041460                JSR     PC,CLDISK       ;CLEAR DISK REG.
                                ;
  014474  012711  003566                MOV     #3566,@R1       ;LOAD RHCS1 WITH ANY NUMBER
  014500  010146                        MOV     R1,-(SP)        ;GETTING READY TO FORM ODD BYTE
  014502  005216                        INC     (SP)            ;SP NOW HAS ODD BYTE FOR RHCS1
  014504  112736  000005                MOVB    #5,@(SP)+       ;MOVE 5 INTO ODD BYTE FOR RHCS1
  014510  011137  001126                MOV     @R1,$BDDAT      ;TEST DATA
```

```
                                                50
014514  022737  004766  001126                  51          CMP     #566!DVA!RDY,$BDDAT ;RHCS1 SHOULD HAVE 4766
014522  001406                                  52          BEQ     1$                  ;BRANCH IF GOOD
014524  012737  004766  001124                  53          MOV     #566!DVA!RDY,$GDDAT ;GOOD DATA
014532  010137  041130                          54          MOV     R1,REGADR           ;FAILING REGISTER RHCS1
014536  104001                                  55          EMT     1
                                                56                                      ;ODD BYTE OF RHCS1 GAVE
                                                57                                      ;WRONG RESULTS
                                                58
014540  112711  000032                          59  1$:     MOVB    #32,@R1             ;MOVE INTO EVEN BYTE
014544  011137  001126                          60          MOV     @R1,$BDDAT          ;TEST DATA
014550  022737  004632  001126                  61          CMP     #432!DVA!RDY,$BDDAT ;RHCS1 SHOULD HAVE 4632
014556  001406                                  62          BEQ     TST36               ;;BRANCH IF GOOD
014560  012737  004632  001124                  63          MOV     #432!DVA!RDY,$GDDAT ;GOOD DATA
014566  010137  041130                          64          MOV     R1,REGADR           ;FAILING REGISTER RHCS1
014572  104001                                  65          EMT     1
                                                66                                      ;BYTE OF RHCS1 GAVE WRONG
                                                67                                      ;RESULTS
                                                68
                                                72
                                                            ;;******************************************************************
                                                            ;*TEST 36       TEST ODD BYTE INSTRUCTION ON RHCS2
                                                            ;*IR (BIT 06) AND THE UNIT SELECT (BIT 0-2) WILL BE SET
                                                            ;;******************************************************************
014574  000004                                  TST36:  SCOPE
014576  012737  000036  001432                  73          MOV     #36,TSTNM           ;MOVE #36 TO TEST NUMBER
                                                74
                                                            ;CHECK TO SEE IF THE PROGRAM IS RUNNING WITH AN RH70

014604  005737  001436                              TST     RH70                ;TEST FLAG FOR RH70 CONTROLLER
014610  001402                                      BEQ     64$                 ;IF FLAG = 1, THIS TEST IS SKIPPED
014612  000137  014752                              JMP     TST37               ;JUMP TO NEXT TEST
014616                                          64$:                            ;IF FLAG = 0, DO THIS TEST
014616  004737  041460                          75      JSR     PC,CLDISK           ;GIVE INIT & SETUP REGISTER CORRES
                                                76
014622  052712  177000                          77      BIS     #177000,(R2)        ;LOAD RHCS2
014626  010246                                  78      MOV     R2,-(SP)            ;GETTING READY FOR ODD BYTE
014630  005216                                  79      INC     (SP)                ;SP NOW HAS ODD BYTE FOR RHCS2
014632  105036                                  80      CLRB    @(SP)+              ;CLERR RHCS2 ODD BYTE
014634  013746  001374                          81      MOV     UNIT,-(SP)          ;GET UNIT NO.
014640  052716  000100                          82      BIS     #IR,(SP)            ;INPUT READY AS IT IS SET
014644  011237  001126                          83      MOV     @R2,$BDDAT          ;TEST DATA
014650  022637  001126                          84      CMP     (SP)+,$BDDAT        ;COMPARE TO SEE THAT
                                                85                                  ;"CLRB" DID CLEAR
014654  001411                                  86      BEQ     1$
014656  013737  001374  001124                  87      MOV     UNIT,$GDDAT
014664  052737  000100  001124                  88      BIS     #IR,$GDDAT          ;GOOD DATA
014672  010237  041130                          89      MOV     R2,REGADR           ;FAILING REGISTER RHCS2
014676  104001                                  90      EMT     1
                                                91                                  ;GAVE WRONG RESULTS
014700  013746  001374                          92  1$: MOV     UNIT,-(SP)
014704  052716  000010                          93      BIS     #BAI,(SP)
014710  052712  020000                          94      BIS     #UPE,@R2            ;HAVE UPE AND MPE IN RHCS2
                                                95                                  ;BESIDES UNIT SELECT
014714  112612                                  96      MOVB    (SP)+,@R2           ;MOVE INTO EVEN BYTE OF RHCS2
014716  013746  001374                          97      MOV     UNIT,-(SP)
014722  052716  020110                          98      BIS     #UPE!IR!BAI,(SP)
014726  011637  001124                          99      MOV     (SP),$GDDAT         ;GOOD DATA
```

```
                                MOV     @R2,$BDDAT      ;TEST DATA
                                CMP     (SP)+,$BDDAT    ;COMPARE TO SEE THAT MOVB DID
                                                        ;MOVE EVEN BYTE ONLY
                                BEQ     TST37           ;;BRANCH IF GOOD
                                MOV     R2,REGADR       ;FAILING REGISTER RHCS2
                                EMT     1
                                                        ;BYTE OF RHCS2 GAVE WRONG
                                                        ;RESULTS


                        ;;**************************************************************
                        ;*TEST 37        ODD BYTE TEST ON RHWC
                        ;*IN THIS REGISTER NO BITS SHOULD BE PERMANENTLY SET
                        ;;**************************************************************
014752  000004          TST37:  SCOPE
                                MOV     #37,TSTNM       ;MOVE #37 TO TEST NUMBER
                                MOV     #STACK,SP       ;RESET STACK
                                JSR     PC,CLDISK       ;CLEAR DISK REGISTERS
                                MOV     RHWC,R4 ;R4 NOW IS WORD COUNT REGISTER
                                MOV     #25252,@R4      ;LOAD RHWC
                                MOV     R4,-(SP)        ;GETTING READY TO FORM ODD BYTE
                                INC     (SP)            ;SP NOW HAS ODD BYTE FOR RHWC
                                MOVB    #377,@(SP)+     ;MOVE 377 INTO ODD BYTE OF RHWC
                                MOV     @R4,$BDDAT      ;TEST DATA
                                CMP     #177652,$BDDAT;COMPARE TO SEE IF MOVB DID OK
                                BEQ     1$              ;BRANCH IF GOOD
                                MOV     #177652,$GDDAT  ;GOOD DATA
                                MOV     R4,REGADR       ;REGISTER FAILING RHWC
                                EMT     1
                                                        ;GAVE WRONG RESULTS
                        1$:     MOVB    #123,@R4        ;MOVE INTO EVEN BYTE OF RHWC
                                MOV     @R4,$BDDAT      ;TEST DATA
                                CMP     #177523,$BDDAT
                                BEQ     TST40           ;;BRANCH IF GOOD
                                MOV     #177523,$GDDAT  ;GOOD DATA
                                MOV     R4,REGADR       ;REGISTER FAILING RHWC
                                EMT     1


                        ;;**************************************************************
                        ;*TEST 40        TEST ODD BYTE INSTRUCTION ON RHBA
                        ;*BIT 0 SHOULD ALWAYS BE 0
                        ;;**************************************************************
015076  000004          TST40:  SCOPE
                                MOV     #STACK,SP       ;RESET STACK
                                MOV     #40,TSTNM       ;MOVE #40 TO TEST NUMBER
                                JSR     PC,CLDISK
                                MOV     RHBA,R4 ;R4 HAS ADDRESS OF RHBA
                                MOV     #25253,@R4      ;LOAD RHBA
                                MOV     R4,-(SP)        ;GETTING READY FOR ODD BYTE
                                INC     (SP)            ;SP HAS ODD BYTE ADR. OF RHBA
                                MOVB    #377,@(SP)+     ;LOAD ODD BYTE OF RHBA
                                MOV     @R4,$BDDAT      ;TEST DATA
                                CMP     #177652,$BDDAT;COMPARE MOVB RESULTS
                                BEQ     1$              ;BRANCH IF GOOD
                                MOV     #177652,$GDDAT;GOOD DATA
                                MOV     R4,REGADR       ;FAILING REGISTER RHBA
                                EMT     1
                                                        ;RHBA GAVE WRONG RESULTS
```

```
    155 015166  112714  000125          1$:     MOVB    #125,@R4
    156 015172  011437  001126                  MOV     @R4,$BDDAT      ;TEST DATA
    157 015176  022737  177524  001126          CMP     #177524,$BDDAT
    158 015204  001406                          BEQ     TST41           ;;BRANCH IF GOOD
    159 015206  012737  177524  001124          MOV     #177524,$GDDAT;GOOD DATA
    160 015214  010437  041130                  MOV     R4,REGADR       ;FAILING REGISTER RHBA
    161 015220  104001                          EMT     1
    162                                                                 ;RHBA GAVE WRONG RESULTS
```

```
                                .SBTTL  DCL COMMAND TESTS

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

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

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

                                ;;*****************************************************************
                                ;*TEST 41        TEST ILF BIT #0 IN REG. RHER1
                                ;*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 TESTED
                                ;;*****************************************************************
015222  000004                  TST41:  SCOPE
015224  012706  001100                  MOV     #STACK,SP       ;RESET STACK
015230  012737  000041  001432          MOV     #41,TSTNM       ;MOVE #41 TO TEST NUMBER
015236  004737  041460                  JSR     PC,CLDISK       ;CLEAR REGISTERS
015242  012777  000001  164010          MOV     #DMD,@RHMR      ;SET DIAGNOSTIC MODE
015250  005037  001422                  CLR     TMPILL  ;GET READY TO MAKE ILLEGAL FUNCTION
015254  012700  001442          1$:     MOV     #FUTABL,R0      ;LOAD FUNCTION CODE TABLE START
015260  012705  000021                  MOV     #17.,R5         ;COUNTER (16 GOOD FUNCTIONS)
015264  023720  001422          2$:     CMP     TMPILL,(R0)+    ;IS THIS A LEGAL FUNCTION CODE?
015270  001004                          BNE     3$              ;NO - DECR. FUNCT. CODE CTR
015272  062737  000002  001422          ADD     #2,TMPILL       ;YES MAKE NEXT FUNCTION CODE
015300  000765                          BR      1$              ;TEST NEXT FUNCTION CODE
015302  005305                  3$:     DEC     R5              ;MAKE NEXT CODE IF 1ST 16
                                                                ;LEGAL FUNCTIONS NOT DONE
015304  001367                          BNE     2$              ;BRANCH IF 16 NOT COMPLETE
015306  032737  000100  001422          BIT     #100,TMPILL     ;ALL BITS UP TO BIT #5 COMPARED?
015314  001077                          BNE     12$             ;YES - EXIT
015316  013737  001422  001504          MOV     TMPILL,ILLEGL;NO - TEST THE ILLEGAL FUNCTION
015324  062737  000002  001422          ADD     #2,TMPILL       ;TEST NEW FUNCTION CODE NEXT TIME

015332  004737  041460          4$:     JSR     PC,CLDISK
015336  012777  000001  163714          MOV     #DMD,@RHMR      ;SET DIAGNOSTIC MODE
015344  013711  001504                  MOV     ILLEGL,@R1      ;ILLEGAL FUNCTION RHCS1
015350  012737  015332  001110          MOV     #4$,$LPERR      ;ERROR RETURN POINT
015356  004737  041064                  JSR     PC,PUTREG       ;SAVE REGISTERS
015362  005737  001316                  TST     ER1             ;THERE SHOULD NOT BE ANY ERROR YET
015366  001403                          BEQ     5$              ;CONTINUE IF RHER1 STILL = 0
015370  010437  001122                  MOV     R4,$BDADR       ;FAILING REGISTER ADDRESS RHER1
015374  104011                          EMT     11
                                                                ;HAS BEEN MOVED INTO RHCS1
                                                                ;NO ERRORS SHOULD SHOW TILL
                                                                ;GO IS SET RHER1 SHOULD BE
                                                                ;ALL ZEROS
```

```
                                5$:     BIS     #GO,@R1         ;GO IN RHCS1
                                        JSR     PC,PUTREG       ;SAVE REGISTERS
                                        CMP     #ILF,ER1        ;ILLEGAL FUNCTION BIT SHOULD BE SET
                                        BEQ     6$              ;IT IS - CONTINUE
                                        MOV     R4,$BDADR       ;FAILING REGISTER ADDRESS RHER1
                                        EMT     11
                                                                ;SET ON AN ILLEGAL FUNCTION
                                                                ;EXECUTION, THE ILLEGAL FUNCTION
                                                                ;BEING EXECUTED IS IN RHCS1

                                6$:     MOV     DS1,-(SP)       ;GET RHDS1
                                        BIC     #PROG,(SP)      ;MASK PROG
                                        CMP     #ATA!ERR!VV!DPR!DRY,(SP)+
                                                                ;ATTENTION (BIT 15)
                                                                ;VOLUME VALID (BIT 6)
                                                                ;COMPOSIT ERROR (BIT 14)
                                                                ;DEVICE READY (BIT 7) SHOULD
                                                                ;BE SET ON RHDS1
                                        BEQ     7$              ;THEY ARE - CONTINUE
                                        MOV     RHDS1,$BDADR ;FAILING REGISTER ADDRESS RHDS1
                                        EMT     11
                                                                ;WITH AN ILLEGAL FUNCTION
                                                                ;ATTENTION (BIT 15)
                                                                ;COMPOSIT ERROR (BIT 14)
                                                                ;MEDIUM ON LINE (BIT 12)
                                                                ;DEVICE READY (BIT 7)

                                7$:     JSR     PC,MIDDLE       ;GIVE A WRITE HEADER AND
                                                                ;DATA COMMAND WITHOUT
                                                                ;CLEARING THE ERRORS
                                                                ;USING 'MIDDLE" SO THAT
                                                                ;IT WILL COME BACK BEFORE
                                                                ;THE END TO FIND OUT ITS
                                                                ;STATE
                                        MOV     R2,10$  ;MOVE RHCS2 ADDRESS
                                        WAT                     ;WAIT FOR 'MXF' BIT
                                10$:    .WORD   0               ;ADDRESS OF RHCS2
                                        MXF
                                11$:    JSR     PC,PUTREG       ;SAVE REGISTERS

                                        BIT     #TRE,CS1        ;TRANSFER ERROR (BIT 14) RHCS1 - 'TRE'
                                                                ;SHOULD SET DUE TO 'MXF'
                                        BNE     13$             ;IT IS - CONTINUE
                                        MOV     R1,$BDADR       ;FAILING REGISTER RHCS1
                                        EMT     11
                                                                ;SHOULD BE SET DUE TO 'MXF'
                                                                ;LOCAL SCOPE RETURN POINT

                                13$:    BR      1$              ;GO BACK & TEST NEXT FUNCTION CODE

                                12$:    NOP

                                ;;*****************************************************************
                                ;*TEST 42        READ IN PRESET
                                ;*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
                                        ;;*****************************************************************
    015516  000004                      TST42:  SCOPE
117 015520  012706  001100              MOV     #STACK,SP       ;RESET STACK
118 015524  012737  000042  001432      MOV     #42,TSTNM       ;MOVE #42 TO TEST NUMBER
119
120 015532  004737  041460              JSR     PC,CLDISK       ;INIT AND SET GENERAL REGISTERS
121
122                                     ;FILL ALL POSSIBLE BITS WITH ONES
123
124 015536  012777  177777  163466      MOV     #177777,@RHWC   ;WORD COUNT REGISTER GETS 177777
125 015544  012777  177777  163462      MOV     #177777,@RHBA   ;BUS ADDRESS REGISTER GETS 177777
126 015552  012777  017437  163464      MOV     #17437,@RHDST   ;DESIRED SECTOR TRACK GETS 17437
127 015560  012777  016377  163462      MOV     #16377,@RHOF    ;OFFSET REGISTER GETS 16277
128 015566  012777  000777  163456      MOV     #777,@RHCA      ;DESIRED CYLINDER GETS 777
129 015574  012746  001400              MOV     #A16!A17,-(SP)  ;GET BIT 9 AND 8
130 015600  053716  001502              BIS     READIN,(SP)
131 015604  012677  163430              MOV     (SP)+,@RHCS1    ;FILL READ IN PRESET IN RHCS1
132 015610  012777  000001  163442      MOV     #DMD,@RHMR      ;SET DIAGNOSTIC MODE
133
134                                     ;THE REGISTERS WILL BE SAVED IN REINTO BUFFER
135 015616  004037  042152              JSR     R0,SAVER        ;SAVE
136 015622  001232                      RHWC                    ;FROM
137 015624  002554                      REINTO                  ;TO
138 015626  000021                      17.                     ;NUMBER SAVED
139
140                                     ;GIVE READ IN PRESET COMMAND
141 015630  052777  000001  163402      BIS     #GO,@RHCS1      ;INCLUDE GO TO READ IN PRESET
142
143                                     ;NOW SAVED REGISTERS WILL BE CHANGED TO EXPECTED VALUE
144 015636  005037  002566              CLR     REINTO+12       ;CLEAR SAVED RHDST
145 015642  042737  016000  002572      BIC     #FMT22!HCI!ECI,REINTO+16 ;CLEAR FMT22,HCI,ECI IN
146                                                             ;SAVED RHOF
147 015650  052737  000100  002572      BIS     #VV,REINTO+16   ;SET VV IN SAVED RHOF
148 015656  005037  002574              CLR     REINTO+20       ;CLEAR SAVED RHCA
149
150                                     ;AFTER A READ IN PRESET COMMAND
151                                     ;SAVE REGISTERS AGAIN SO THAT COMPARES CAN BE DONE
152
153 015662  004037  042152              JSR     R0,SAVER        ;SAVE
154 015666  001232                      RHWC                    ;FROM
155 015670  001510                      WRFROM                  ;TO
156 015672  000021                      17.                     ;NUMBER OF REGISTERS SAVED
157
158                                     ;AS UPPER BYTE OF RHAS CAN BE CHANGING IN A DUAL PORT
159                                     ;OPERATION THE UPPER BYTE OF RHAS WILL BE SAVED AS IS
160                                     ;SO THAT THE COMPARES ARE ONLY VALID FOR THE LOWER BYTE
161 015674  113737  002601  001535      MOVB    REINTO+25,WRFROM+25;SAVE UPPER RHAS
162
163                                     ;COMPARE REGISTERS BEFORE READ IN PRESET COMMAND
164                                     ;WITH AFTER COMMAND
165
166 015702  004037  042354              JSR     R0,COMPAR       ;COMPARE
167 015706  002554                      REINTO                  ;GOOD BUFFER
168 015710  001510                      WRFROM                  ;TEST BUFFER
169 015712  000021                      17.                     ;NUMBER OF REGISTERS
```

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

                        046426          1$:     MOV     ERWORD,R5       ;GETTING READY TO INDEX
                                                ADD     R5,R5           ;DOUBLE ERROR WORD
                        001230  041130          MOV     RHWC-2(R5),REGADR ;FAILING REG. ADDRESS
                                                EMT     1
                                                                        ;REGISTER CHANGE
                                                RTS     PC              ;RETURN FOR FURTHER COMPARISONS

                                2$:                                     ;NO ERRORS

                                ;;*****************************************************************
                                ;*TEST 43       NO OPERATION FUNCTION TEST
                                ;*ALL POSSIBLE REGISTERS ARE CLEARED THEN A'NOP''=C
                                ;*IS GIVEN NO CHANGE SHOULD HAPPEN
                                ;*ALL POSSIBLE REGISTERS ARE FILLED WITH ONES THEN A 'NOP''
                                ;*IS GIVEN NO CHANGE SHOULD HAPPEN
                                ;;*****************************************************************
                                TST43:  SCOPE
                        000043  001432          MOV     #43,TSTNM       ;MOVE #43 TO TEST NUMBER

                                ;START WITH CLR IN RHCS2 (BIT5)

                        041460                  JSR     PC,CLDISK       ;CLEAR ALL POSSIBLE BITS
                        000001  163274          MOV     #DMD,@RHMR      ;SET DIAGNOSTIC MODE
                        001442                  MOV     NOPERA,@R1      ;PUT NOP OPERATION=0 IN RHCS1
                        001232                  MOV     #RHWC,R0        ;STARTING ADDRESS OF REG
                        001306                  MOV     #WC,R3          ;STARTING ADDRESS OF WHERE SAVED
                        000021                  MOV     #RHEC2-RHWC+2/2,R2 ;NUMBER OF REGISTERS
                                1$:             MOV     @(R0)+,(R3)+    ;SAVE HARDWARE REG
                                                DEC     R2              ;COUNT
                                                BNE     1$              ;BRANCH IF NOT COMPLETE
                        001262  016032          MOV     RHDS1,2$        ;GET ADDRESS OF DRIVE STATUS
                        016040                  MOV     R1,3$           ;GET ADDRESS OF RHCS1
                        000001                  BIS     #GO,@R1         ;GO TO RHCS1
                                                WAT                     ;WAIT FOR DRY IN RHDS1
                                2$:             .WORD   0               ;ADDRESS OF DRIVE STATUS RHDS1
                                                DRY                     ;DRY WILL BE WAITED ON
                                                WAT                     ;WAIT FOR RDY IN RHCS1
                                3$:             .WORD   0               ;ADDRESS OF RHCS1 PUT HERE BY AN
                                                                        ;EARLIER MOV
                                                RDY                     ;RDY WILL BE WAITED ON

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

                        042152                  JSR     R0,SAVER        ;SAVE
                                                RHWC                    ;FROM
                                                WRFROM                  ;TO
                                                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
                        001333  001535          MOVB    AS+1,WRFROM+25;SAVE UPPER RHAS
```

```
                                        ;COMPARE REGISTERS BEFORE  NO OP COMMAND
                                        ;WITH AFTER COMMAND
016064 004037 042354                    JSR     R0,COMPAR       ;COMPARE
016070 001306                           WC                      ;GOOD BUFFER
016072 001510                           WRFROM                  ;TEST BUFFER
016074 000021                           17.                     ;NUMBER OF REGISTERS
016076 016104                           4$                      ;RETURN FOR ERROR
016100 016104                           4$                      ;SAME
016102 016124                           5$                      ;RETURN FOR GOOD COMPARISON

016104 013705 046426            4$:     MOV     ERWORD,R5       ;GETTING READY TO INDEX
016110 060505                           ADD     R5,R5           ;DOUBLE ERROR WORD
016112 016537 001230 041130             MOV     RHWC-2(R5),REGADR ;FAILING REG. ADDRESS
016120 104001                           EMT     1
                                                                ;REGISTER CHANGE
016122 000207                           RTS     PC              ;RETURN FOR FURTHER COMPARISONS

016124                          5$:                             ;NO ERRORS


016124 012737 016132 001110             MOV     #14$,$LPERR     ;SET SCOPE LOOP TO 14$
016132 004737 041460            14$:    JSR     PC,CLDISK       ;INIT LAST ALL ZERO TEST
016136 012777 000001 163114             MOV     #DMD,@RHMR      ;SET DIAGNOSTIC MODE


                                        ;NOW START WITH ALL ONES IN ALL POSSIBLE REGISTERS

016144 012700 001232                    MOV     #RHWC,R0        ;ADDRESS OF FIRST REGISTER
016150 012705 000021                    MOV     #RHEC2-RHWC+2/2,R5 ;NO. OF REGISTERS
016154 012730 177676            6$:     MOV     #177676,@(R0)+  ;FILL WITH ALL ONES
016160 013777 001374 163050             MOV     UNIT,@RHCS2     ;REINSTATE UNIT NUMBER UNDER TEST
                                                                ;KEEP INTERRUPT DISABLED
016166 005305                           DEC     R5              ;COUNT
016170 001371                           BNE     6$              ;BRANCH IF INCOMPLETE
016172 013711 001442                    MOV     NOPERA,@R1      ;PUT NOP OPERATION =0 IN RHCS1
016176 01270C 001232                    MOV     #RHWC,R0        ;STARTING ADDRESS OF REG
016202 012703 001306                    MOV     #WC,R3          ;STARTING ADDRESS OF WHERE SAVED
016206 012702 000021                    MOV     #RHEC2-RHWC+2/2,R2 ;NUMBER OF REGISTERS
016212 013023                   7$:     MOV     @(R0)+,(R3)+    ;SAVE HARDWARE REG
016214 005302                           DEC     R2              ;COUNT
016216 001375                           BNE     7$              ;BRANCH IF NOT COMPLETE
016220 013737 001262 016240             MOV     RHDS1,10$       ;GET ADDRESS OF DRIVE STATUS
016226 010137 016246                    MOV     R1,11$  ;GET ADDRESS OF RHCS1
016232 052711 000001                    BIS     #GO,@R1         ;GO TO RHCS1
016236 104415                           WAT                     ;WAIT FOR DRY IN RHDS1
016240 000000                   10$:    .WORD   0               ;ADDRESS OF DRIVE STATUS RHDS1
016242 000200                           DRY                     ;DRY WILL BE WAITED ON
016244 104415                           WAT                     ;WAIT FOR RDY IN RHCS1
016246 000000                   11$:    .WORD   0               ;ADDRESS OF RHCS1 PUT HERE BY AN
                                                                ;EARLIER MOV.
016250 000200                           RDY                     ;RDY WILL BE WAITED ON

                                        ;AFTER A NO OP COMMAND
```

```
                                        ;SAVE REGISTERS AGAIN SO THAT COMPARES CAN BE DONE

016252  004037  042152                        JSR     R0,SAVER        ;SAVE
016256  001232                                RHWC                    ;FROM
016260  001510                                WRFROM                  ;TO
016262  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
016264  113737  001333  001535                MOVB    AS+1,WRFROM+25;SAVE UPPER RHAS


                                        ;COMPARE REGISTERS BEFORE NO OP COMMAND
                                        ;WITH AFTER COMMAND

016272  004037  042354                        JSR     R0,COMPAR       ;COMPARE
016276  001306                                WC                      ;GOOD BUFFER
016300  001510                                WRFROM                  ;TEST BUFFER
016302  000021                                17.                     ;NUMBER OF REGISTERS
016304  016312                                12$                     ;RETURN FOR ERROR
016306  016312                                12$                     ;SAME
016310  016332                                13$                     ;RETURN FOR GOOD COMPARISON

016312  013705  046426          12$:          MOV     ERWORD,R5       ;GETTING READY TO INDEX
016316  060505                                ADD     R5,R5           ;DOUBLE ERROR WORD
016320  016537  001230  041130                MOV     RHWC-2(R5),REGADR ;FAILING REG. ADDRESS
016326  104001                                EMT     1
                                                                      ;REGISTER CHANGE
016330  000207                                RTS     PC              ;RETURN FOR FURTHER COMPARISONS

016332                                  13$:                          ;NO ERRORS

                                        ;;*****************************************************************
                                        ;*TEST 44        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
                                        ;;*****************************************************************
016332  000004                          TST44:  SCOPE
016334  012706  001100                        MOV     #STACK,SP       ;RESET STACK
016340  012737  000044  001432                MOV     #44,TSTNM       ;MOVE #44 TO TEST NUMBER
016346  004737  041460                        JSR     PC,CLDISK       ;SET REGISTERS AND CLEAR

                                        ;FILL ALL POSSIBLE BITS WITH ONES

016352  012777  177777  162650                MOV     #177777,@RHDB   ;BUS ADDRESS REGISTER GETS 177777
016360  012777  177777  162644                MOV     #177777,@RHWC   ;WORD COUNT REGISTER GETS 177777
016366  012777  177777  162640                MOV     #177777,@RHBA   ;BUS ADDRESS REGISTER GETS 177777
016374  052777  157010  162634                BIS     #157010,@RHCS2  ;CONTROL AND STATUS 2 GETS 157010
016402  012777  001476  162630                MOV     #1476,@RHCS1    ;CONTROL AND STATUS REGISTER/GETS 1476
016410  012777  177777  162624                MOV     #177777,@RHER1  ;ERROR REGISTER1 GETS 177777
016416  012777  017437  162620                MOV     #17437,@RHDST   ;DESIRED SECTOR TRACK
016424  012777  177777  162614                MOV     #177777,@RHER2  ;ERROR REGISTER 2
016432  012777  016277  162610                MOV     #16277,@RHOF    ;OFFSET REGISTER
```

```
                                                ;DESIRED CYLINDER
                                                ;ERROR REGISTER 3
                                                ;MAINTENANCE REGISTER
                                                ;MAINTENANCE REGISTER


                        ;THIS SETS BITS FOR ALL PRESENT DRIVES

                                                ;GET DRIVE PRESENT
                                                ;CLEAR RHCS2 AND CARRY BIT
                                                ;COUNTER
                        30$:                    ;GET BIT INTO CARRY
                                                ;BRANCH IF NO UNIT ON THIS BIT
                                                ;MOVE INTO ERROR REGISTER TO SET ATA
                        31$:                    ;INCREMENT RHCS2 - UNIT NO.
                                                ;COUNT
                                                ;BRANCH IF 8 DONE
                                                ;CONTINUE THIS ROUTINE
                        27$:
                                                ;REINSTATE SET BITS
                                                ;


                                                ;SET DMD
                                                ;DRIVE CLEAR = 10 INTO RHCS1
                                                ;GO
                                                ;R0 CONTAINS ADDR. OF ADDR. OF REG.


                        ;DATA BUFFER REGISTER


016556 012737 177777 001124 28$:                ;GOOD DATA FOR ERROR TYPEOUT
016564 013037 001126                            ;TEST DATA
016570 022737 177777 001126                     ;COMPARE DATA
016576 001402                                   ;BRANCH IF GOOD
016600 004737 017436                            ;JUMP TO ERROR FOR CLR (BIT 5)
                                                ;IN RHCS2


                        ;WORD COUNT REGISTER


016604 012737 177777 001124 3$:                 ;GOOD DATA FOR ERROR TYPEOUT
016612 013037 001126                            ;TEST DATA
016616 022737 177777 001126                     ;COMPARE DATA
016624 001402                                   ;BRANCH IF GOOD
016626 004737 017436                            ;JUMP TO ERROR FOR CLR (BIT 5)
                                                ;IN RHCS2


                        ;BUS ADDRESS REGISTER


016632 012737 177776 001124 4$:                 ;GOOD DATA FOR ERROR TYPEOUT
016640 013037 001126                            ;TEST DATA
016644 022737 177776 001126                     ;COMPARE DATA
```

```
016652  001402                          BEQ     5$              ;BRANCH IF GOOD
016654  004737  017436                  JSR     PC,ERCLFC       ;JUMP TO ERROR FOR CLR (BIT 5)
                                                                ;IN RHCS2

                        ;CONTROL AND STATUS 2 REGISTER


016660  012746  000110          5$:     MOV     #110,-(SP)      ;INCLUDE IR
016664  053716  001374                  BIS     UNIT,(SP)       ;SET UNIT NO.
016670  012637  001124                  MOV     (SP)+,$GDDAT    ;GOOD DATA FOR TYPE OUT
016674  013037  001126                  MOV     @(R0)+,$BDDAT   ;TEST DATA
016700  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE DATA
016706  001402                          BEQ     6$              ;BRANCH IF GOOD
016710  004737  017436                  JSR     PC,ERCLFC       ;JUMP TO ERROR FOR CLR (BIT 5)
                                                                ;IN RHCS2

                        ;CONTROL AND STATUS 1 REGISTER

016714  005737  001400          6$:     TST     NUNIT           ;ARE THERE MORE THAN ONE UNIT
016720  001404                          BEQ     32$             ;BRANCH IF ONLY ONE UNIT
016722  012737  104210  001124          MOV     #104210,$GDDAT;GOOD DATA
016730  000403                          BR      33$
016732  012737  004210  001124  32$:    MOV     #4210,$GDDAT    ;GOOD DATA
016740  013037  001126          33$:    MOV     @(R0)+,$BDDAT   ;TEST DATA

016744  023737  001124  001126          CMP     $GDDAT,$BDDAT;COMPARE DATA
016752  001402                          BEQ     7$              ;BRANCH IF GOOD
016754  004737  017436                  JSR     PC,ERCLFC       ;JUMP TO ERROR FOR CLR BIT 5
                                                                ;IN RHCS2

                        ;ERROR 1 REGISTER


016760  012737  000000  001124  7$:     MOV     #0,$GDDAT       ;GOOD DATA FOR ERROR TYPEOUT
016766  013037  001126                  MOV     @(R0)+,$BDDAT   ;TEST DATA
016772  022737  000000  001126          CMP     #0,$BDDAT       ;COMPARE DATA
017000  001402                          BEQ     10$             ;BRANCH IF GOOD
017002  004737  017436                  JSR     PC,ERCLFC       ;JUMP TO ERROR FOR CLR (BIT 5)
                                                                ;IN RHCS2

                        ;DESIRED SECTOR/TRACK REGISTER


017006  012737  017437  001124  10$:    MOV     #17437,$GDDAT   ;GOOD DATA FOR ERROR TYPEOUT
017014  013037  001126                  MOV     @(R0)+,$BDDAT   ;TEST DATA
017020  022737  017437  001126          CMP     #17437,$BDDAT   ;COMPARE DATA
017026  001402                          BEQ     11$             ;BRANCH IF GOOD
017030  004737  017436                  JSR     PC,ERCLFC       ;JUMP TO ERROR FOR CLR (BIT 5)
                                                                ;IN RHCS2

                        ;ERROR 2 REGISTER


017034  012737  000000  001124  11$:    MOV     #0,$GDDAT       ;GOOD DATA FOR ERROR TYPEOUT
```

```
017042  013037  001126                  MOV     @(R0)+,$BDDAT   ;TEST DATA
017046  022737  000000  001126          CMP     #0,$BDDAT       ;COMPARE DATA
017054  001402                          BEQ     12$             ;BRANCH IF GOOD
017056  004737  017436                  JSR     PC,ERCLFC       ;JUMP TO ERROR FOR CLR (BIT 5)
                                                                ;IN RHCS2
414
415                             ;OFFSET REGISTER
416
417
017062  012737  116000  001124  12$:    MOV     #116000,$GDDAT  ;GOOD DATA FOR ERROR TYPEOUT
017070  013037  001126                  MOV     @(R0)+,$BDDAT   ;TEST DATA
017074  022737  116000  001126          CMP     #116000,$BDDAT  ;COMPARE DATA
017102  001402                          BEQ     13$             ;BRANCH IF GOOD
017104  004737  017436                  JSR     PC,ERCLFC       ;JUMP TO ERROR FOR CLR (BIT 5)
                                                                ;IN RHCS2
418
419                             ;DESIRED CYLINDER ADDRESS REGISTER
420
421
017110  012737  001777  001124  13$:    MOV     #1777,$GDDAT    ;GOOD DATA FOR ERROR TYPEOUT
017116  013037  001126                  MOV     @(R0)+,$BDDAT   ;TEST DATA
017122  022737  001777  001126          CMP     #1777,$BDDAT    ;COMPARE DATA
017130  001402                          BEQ     14$             ;BRANCH IF GOOD
017132  004737  017436                  JSR     PC,ERCLFC       ;JUMP TO ERROR FOR CLR (BIT 5)
                                                                ;IN RHCS2
422
423                             ;ERROR 3 REGISTER
424
425
017136  012737  000000  001124  14$:    MOV     #0,$GDDAT       ;GOOD DATA FOR ERROR TYPEOUT
017144  013037  001126                  MOV     @(R0)+,$BDDAT   ;TEST DATA
017150  022737  000000  001126          CMP     #0,$BDDAT       ;COMPARE DATA
017156  001402                          BEQ     15$             ;BRANCH IF GOOD
017160  004737  017436                  JSR     PC,ERCLFC       ;JUMP TO ERROR FOR CLR (BIT 5)
                                                                ;IN RHCS2
426
427                             ;ATTENTION SUMMARY REGISTER
428
429 017164  013737  001420  001124  15$:    MOV     TOTALAT,$GDDAT;SET ALL BITS OF DRIVE PRESENT IN RHAS
430 017172  043737  001416  001124          BIC     ATTENT,$GDDAT   ;CLEAR ONLY WORKING DRIVE BIT
431 017200  013037  001126                  MOV     @(R0)+,$BDDAT           ;GET RHAS
432 017204  123737  001124  001126          CMPB    $GDDAT,$BDDAT   ;COMPARE DATA
433 017212  001402                          BEQ     16$                     ;BRANCH IF GOOD
434 017214  004737  017436                  JSR     PC,ERCLFC               ;JUMP TO ERROR FOR CLR (BIT 5) IN RHCS2
435
436                             ;MAINTAINABILITY REGISTER
437
438
017220  012737  000400  001124  16$:    MOV     #400,$GDDAT     ;GOOD DATA FOR ERROR TYPEOUT
017226  013037  001126                  MOV     @(R0)+,$BDDAT   ;TEST DATA
017232  022737  000400  001126          CMP     #400,$BDDAT     ;COMPARE DATA
017240  001402                          BEQ     17$             ;BRANCH IF GOOD
017242  004737  017436                  JSR     PC,ERCLFC       ;JUMP TO ERROR FOR CLR (BIT 5)
                                                                ;IN RHCS2
439
440                             ;DRIVE STATUS REGISTER
441
```

```
                                                                ;GOOD DATA FOR PRINTOUT
017246  012737  000700  001124  17$:    MOV     #700,$GDDAT    ;GET RHDS1
017254  013046                         MOV     @(R0)+,-(SP)   ;TEST DATA
017256  011637  001126                 MOV     (SP),$BDDAT    ;CLEAR PROG BIT
017262  042716  001000                 BIC     #PROG,(SP)     ;COMPARE DATA
017266  022726  000700                 CMP     #700,(SP)+     ;BRANCH IF GOOD
017272  001402                         BEQ     20$            ;JUMP TO ERROR FOR DRIVE CLEAR
017274  004737  017436                 JSR     PC,ERCLFC
                                                               ;DRIVE TYPE
                                                               ;GOOD DATA FOR ERROR TYPEOUT
017300  013737  001410  001124  20$:    MOV     SAVDT,$GDDAT   ;TEST DATA
017306  013037  001126                 MOV     @(R0)+,$BDDAT  ;COMPARE DATA
017312  023737  001410  001126         CMP     SAVDT,$BDDAT   ;BRANCH IF GOOD
017320  001402                         BEQ     21$            ;JUMP TO ERROR FOR CLR (BIT 5)
017322  004737  017436                 JSR     PC,ERCLFC      ;IN RHCS2
                                                               ;SERIAL NUMBER REGISTER
                                                               ;GOOD DATA FOR ERROR TYPEOUT
017326  013737  001412  001124  21$:    MOV     SAVSN,$GDDAT   ;TEST DATA
017334  013037  001126                 MOV     @(R0)+,$BDDAT  ;COMPARE DATA
017340  023737  001412  001126         CMP     SAVSN,$BDDAT   ;BRANCH IF GOOD
017346  001402                         BEQ     22$            ;JUMP TO ERROR FOR CLR (BIT 5)
017350  004737  017436                 JSR     PC,ERCLFC      ;IN RHCS2
                                                               ;ECC1 POSITION
                                                               ;GOOD DATA FOR ERROR TYPEOUT
017354  012737  000000  001124  22$:    MOV     #0,$GDDAT      ;TEST DATA
017362  013037  001126                 MOV     @(R0)+,$BDDAT  ;COMPARE DATA
017366  022737  000000  001126         CMP     #0,$BDDAT      ;BRANCH IF GOOD
017374  001402                         BEQ     23$            ;JUMP TO ERROR FOR CLR (BIT 5)
017376  004737  017436                 JSR     PC,ERCLFC      ;IN RHCS2
                                                               ;ECC2 PATTERN
                                                               ;GOOD DATA FOR ERROR TYPEOUT
017402  012737  000000  001124  23$:    MOV     #0,$GDDAT      ;TEST DATA
017410  013037  001126                 MOV     @(R0)+,$BDDAT  ;COMPARE DATA
017414  022737  000000  001126         CMP     #0,$BDDAT      ;BRANCH IF GOOD
017422  001402                         BEQ     24$            ;JUMP TO ERROR FOR CLR (BIT 5)
017424  004737  017436                 JSR     PC,ERCLFC      ;IN RHCS2
                                                               ;LOOK-AHEAD REGISTER
                                                               ;AS THE LOOK-AHEAD REG. CANNOT BE PREDICTED
017430  005720          24$:    TST     (R0)+                  ;IT IS NOT CHECKED AFTER AN INIT
                                                               ;CURRENT CYLINDER ADDRESS REGISTER
```

```
                                25$:    TST     (R0)+           ;AS THE CURRENT CYL REG. CANNOT BE PREDICTED
                                                                ;AFTER AN INIT IT IS NOT CHECKED
                                26$:
017434  000413                        BR      TST45           ;BRANCH OVER JSR
017436  014037  041130        ERCLFC: MOV     -(R0),REGADR    ;FAILING REGISTER ADDRESS
017442  104001                        EMT     1
                                                                ;NOT CLEAR APPROPIATE BITS
                                                                ;OR CLEARED EXTRA BITS
017444  005720                        TST     (R0)+           ;UNDO -(R0) FOR ERROR
017446  000207                        RTS     PC              ;RETURN TO ABOVE PROGRAM
                                                                ;OR CLEARED EXTRA BITS
017450  005720                        TST     (R0)+           ;UNDO -(R0) FOR BAD DATA
017452  004737  041514                JSR     PC,CHECKT       ;CHECK THAT DVA,RDY,DPR,DRY = 1
017456  104401  062726                TYPE    ,CPHALT         ;AND THAT NO OTHERS = 1.  CANNOT CON-
                                                                ;TINUE TESTING IF BOTH AREN'T TRUE
017462  000000                        HALT                    ;STOP THE TEST

                                ;;*****************************************************************
                                ;*TEST 45       SEEK COMMAND TEST
                                ;*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
                                ;;*****************************************************************
017464  000004                TST45:  SCOPE
017466  012706  001100                MOV     #STACK,SP       ;RESET STACK
017472  012737  000045  001432        MOV     #45,TSTNM       ;MOVE #45 TO TEST NUMBER
017500  004737  041460                JSR     PC,CLDISK       ;INIT AND SET UP GENERAL REG. CORRES.
                                                                ;AND UNIT NUMBER
017504  012777  000001  161546        MOV     #DMD,@RHMR      ;SET DIAGNOSTIC MODE BIT
                                                                ;THIS ENABLES COMMANDS WITHOUT MOL
                                                                ;AND HOLDS RHLA FROM MOVING
017512  005077  161526                CLR     @RHDST          ;MAKE DESIRED SECTOR TRACK LEGAL
017516  013777  001472  161514        MOV     SEECOM,@RHCS1   ;LOAD SEEK COMMAND INTO CONTROLLER
017524  017746  161546                MOV     @RHCC,-(SP)     ;GET CURRENT CYLINDER


                                ;FOLLOWING ARE TWO BLOCKS OF CODE TO LOAD RHCA WITH THE PROPER
                                ;ADDRESS DEPENDING UPON WHETHER THE DRIVE IS AN RP06 OR RP04

017530  005737  001434                TST     RP06    ;MOVE DRIVE TYPE FLAG TO ITSELF TO TEST
017534  001416                        BEQ     11$             ;TREAT THE DRIVE AS AN RP04

                                                                ;TREAT THE DRIVE AS AN RP06
017536  022726  001456                CMP     #814.,(SP)+     ;IS CURRENT CYLINDER SAME AS 814. ?
017542  001404                        BEQ     9$              ;BRANCH IF YES TO MAKE RHCA = 813.
017544  012737  001456  001210        MOV     #814.,$TMP5     ;GET READY TO MAKE RHCA = 814.
017552  000403                        BR      10$             ;FILL RHCA
017554  012737  001455  001210  9$:   MOV     #813.,$TMP5     ;GET READY TO MAKE RHCA = 813.
017562  013777  001210  161462  10$:  MOV     $TMP5,@RHCA     ;MAKE DESIRED CYLINDER 814., OR 813.
017570  000415                        BR      14$             ;SAVE REGISTERS

                                                                ;TREAT THE DRIVE AS AN RP04
017572  022726  000632        11$:    CMP     #410.,(SP)+     ;IS CURRENT CYLINDER SAME AS 410. ?
017576  001404                        BEQ     12$             ;BRANCH IF YES TO MAKE RHCA = 409.
```

```
017600  012737  000632  001210          MOV     #410.,$TMP5     ;GET READY TO MAKE RHCA = 410.
017606  000403                          BR      13$             ;FILL RHCA
017610  012737  000631  001210  12$:    MOV     #409.,$TMP5     ;GET READY TO MAKE RHCA = 409.
017616  013777  001210  161426  13$:    MOV     $TMP5,@RHCA     ;MAKE DESIRED CYLINDER 410., OR 409.

                                ;SAVE REGISTERS FOR COMPARISON AFTER GO

017624  004037  042152          14$:    JSR     R0,SAVER        ;SAVE
017630  001232                          RHWC                    ;FROM
017632  002554                          REINTO                  ;TO
017634  000023                          19.                     ;NUMBER OF REGISTERS SAVED

                                ;GIVE GO TO COMMAND
017636  052777  000001  161374          BIS     #GO,@RHCS1      ;GO TO COMMAND

                                ;CHANGE SAVED REGISTERS TO EXPECTED VALUES

017644  052737  000001  002562          BIS     #GO,REINTO+6    ;SAVED RHCS1
017652  052737  020000  002604          BIS     #PIP,REINTO+30 ;SAVED RHDS1
017660  042737  000200  002604          BIC     #DRY,REINTO+30 ;SAVED RHDS1

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

017666  004037  042152          JSR     R0,SAVER        ;SAVE
017672  001232                  RHWC                    ;FROM
017674  001510                  WRFROM                  ;TO
017676  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

017700  113737  002601  001535          MOVB    REINTO+25,WRFROM+25;SAVE UPPER RHAS

                                ;COMPARE REGISTERS BEFORE SEEK COMMAND
                                ;WITH CONTENTS AFTER GO IS ISSUED

017706  004037  042354          JSR     R0,COMPAR       ;COMPARE
017712  002554                  REINTO                  ;GOOD BUFFER
017714  001510                  WRFROM                  ;TEST BUFFER
017716  000023                  19.                     ;NUMBER
017720  017726                  1$                      ;RETURN FOR ERROR
017722  017726                  1$                      ;SAME
017724  017746                  2$                      ;RETURN FOR GOOD COMPARISON

017726  013705  046426  1$:     MOV     ERWORD,R5       ;GETTING READY TO INDEX
017732  060505                  ADD     R5,R5           ;DOUBLE ERROR WORD
017734  016537  001230  041130  MOV     RHWC-2(R5),REGADR ;FAILING REGISTER ADDRESS

017742  104001                  EMT     1
                                                        ;AFTER SEEK COMMAND
                                                        ;WITH GO IS GIVEN
017744  000207                  RTS     PC              ;RETURN TO COMPARISON
```

```
                                        ;NOW GIVE INIT AND GET GO AND PIP DOWN

017746  052712  000040          2$:     BIS     #CLR,@R2        ;RH INITILIZE
017752  013712  001374                  MOV     UNIT,@R2        ;REINSTATE UNIT NUMBER
017756  012777  000001  161274          MOV     #DMD,@RHMR      ;SET DIAGNOSTIC MODE BIT
                                                                ;THIS ENABLES COMMANDS WITHOUT MOL
                                                                ;AND HOLDS RHLA FROM MOVING

                                        ;CHANGE REGISTERS TO EXPECTED VALUE

017764  042737  000001  002562          BIC     #GO,REINTO+6    ;SAVED RHCS1
017772  042737  020000  002604          BIC     #PIP,REINTO+30 ;SAVED RHDS1
020000  052737  000200  002604          BIS     #DRY,REINTO+30 ;SAVED RHDS1
020006  017737  161262  002616          MOV     @RHLA,REINTO+42;SAVED RHLA
020014  013737  001210  002620          MOV     $TMP5,REINTO+44 ;SAVED RHCC

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

020022  004037  042152                  JSR     R0,SAVER        ;SAVE
020026  001232                          RHWC                    ;FROM
020030  001510                          WRFROM                  ;TO
020032  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

020034  113737  002601  001535          MOVB    REINTO+25,WRFROM+25;SAVE UPPER RHAS

                                        ;COMPARE REGISTERS AFTER INITIALIZE

020042  004037  042354                  JSR     R0,COMPAR       ;COMPARE
020046  002554                          REINTO                  ;GOOD BUFFER
020050  001510                          WRFROM                  ;TEST BUFFER
020052  000023                          19.                     ;NUMBER OF REGISTERS TO BE
                                                                ;COMPARED
020054  020062                          3$                      ;RETURN POINT FOR ERROR
020056  020062                          3$                      ;SAME
020060  020102                          4$                      ;RETURN POINT FOR GOOD COMPARISON

020062  013705  046426          3$:     MOV     ERWORD,R5       ;GETTING READY TO INDEX
020066  060505                  ADD     R5,R5                   ;DOUBLE ERROR WORD
020070  016537  001230  041130          MOV     RHWC-2(R5),REGADR ;FAILING REGISTER ADDRESS
020076  104001                          EMT     1
                                                                ;CONTENTS AFTER GIVING AN
                                                                ;INITILIZE FOLLOWING A
                                                                ;SEEK COMMAND
020100  000207                          RTS     PC              ;RETURN TO COMPARISON

020102                          4$:                             ;GOOD COMPARISON

020102  004737  041460                  JSR     PC,CLDISK       ;INIT AND SET UP GENERAL REG.
```

```
                                                        ;AND UNIT NUMBER
020106  012777  000001  161144          MOV     #DMD,@RHMR      ;SET DIAGNOSTIC MODE BIT
                                                        ;THIS ENABLES COMMANDS WITHOUT MOL
                                                        ;AND HOLDS RHLA FROM MOVING

020114  013777  001472  161116          MOV     SEECOM,@RHCS1   ;LOAD SEEK COMMAND INTO RHC
020122  005077  161124                  CLR     @RHCA           ;DESIRED CYLINDER ADDRESS

                        ;SAVE REGISTERS FOR COMPARISON AFTER GO
020126  004037  042152          JSR     R0,SAVER        ;SAVE
020132  001232                  RHWC                    ;FROM
020134  002554                  REINTO                  ;TO
020136  000023                  19.                     ;NUMBER OF REGISTERS SAVED

                        ;GIVE GO TO SEEK COMMAND
020140  052777  000001  161072          BIS     #GO,@RHCS1      ;GO TO SEEK COMMAND

                        ;CHANGE SAVED REGISTERS TO EXPECTED VALUES

020146  052737  000001  002562          BIS     #GO,REINTO+6    ;SAVED RHCS1
020154  052737  020000  002604          BIS     #PIP,REINTO+30 ;SAVED RHDS1
020162  042737  000200  002604          BIC     #DRY,REINTO+30 ;SAVED RHDS1


                        ;AFTER GO HAS BEEN GIVEN TO SEEK COMMAND
                        ;SAVE REGISTERS AGAIN SO THAT COMPARISONS CAN
                        ;BE DONE
020170  004037  042152          JSR     R0,SAVER        ;SAVE
020174  001232                  RHWC                    ;FROM
020176  001510                  WRFROM                  ;TO
020200  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
020202  113737  002601  001535          MOVB    REINTO+25,WRFROM+25;SAVE UPPER RHAS


                        ;COMPARE REGISTERS BEFORE COMMAND
                        ;WITH CONTENTS AFTER GO IS GIVEN

020210  004037  042354          JSR     R0,COMPAR       ;COMPARE
020214  002554                  REINTO                  ;GOOD BUFFER
020216  001510                  WRFROM                  ;TEST BUFFER
020220  000023                  19.                     ;NUMBER
020222  020230                  5$                      ;RETURN FOR ERROR
020224  020230                  5$                      ;SAME
020226  020250                  6$                      ;RETURN FOR GOOD COMPARISON
020230  013705  046426  5$:     MOV     ERWORD,R5       ;GETTING READY TO INDEX
020234  060505                  ADD     R5,R5           ;DOUBLG ERROR WORD
020236  016537  001230  041130          MOV     RHWC-2(R5),REGADR ;FAILING REGISTER ADDRESS
020244  104001                  EMT     1
                                                        ;AFTER COMMAND
                                                        ;WITH GO IS GIVEN
020246  000207                  RTS     PC              ;RETURN TO COMPARISON

```

```
                                ;NOW GIVE INIT AND GET GO AND PIP DOWN

020250  052712  000040          6$:     BIS     #CLR,@R2        ;RH INITILIZE
020254  013712  001374                  MOV     UNIT,@R2        ;REINSTATE UNIT NUMBER
020260  012777  000001  160772          MOV     #DMD,@RHMR      ;SET DIAGNOSTIC MODE BIT
                                                                ;THIS ENABLES COMMANDS WITHOUT MOL
                                                                ;AND HOLDS RHLA FROM MOVING

                                ;CHANGE REGISTERS TO EXPECTED VALUE

020266  042737  000001  002562          BIC     #GO,REINTO+6    ;SAVED RHCS1
020274  042737  020000  002604          BIC     #PIP,REINTO+30 ;SAVED RHDS1
020302  052737  000200  002604          BIS     #DRY,REINTO+30 ;SAVED RHDS1
020310  017737  160760  002616          MOV     @RHLA,REINTO+42;SAVED RHLA
020316  005037  002620                  CLR     REINTO+44       ;SAVED RHCC


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

020322  004037  042152                  JSR     R0,SAVER        ;SAVE
020326  001232                          RHWC                    ;FROM
020330  001510                          WRFROM                  ;TO
020332  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
020334  113737  002601  001535          MOVB    REINTO+25,WRFROM+25;SAVE UPPER RHAS


                                ;COMPARE REGISTERS AFTER INITALIZE

020342  004037  042354                  JSR     R0,COMPAR       ;COMPARE
020346  002554                          REINTO                  ;GOOD BUFFER
020350  001510                          WRFROM                  ;TEST BUFFER
020352  000023                          19.                     ;NUMBER OF REGISTERS TO BE
                                                                ;COMPARED
020354  020362                          7$                      ;RETURN POINT FOR ERROR
020356  020362                          7$                      ;SAME
020360  020402                          8$                      ;RETURN POINT FOR GOOD COMPARISON

020362  013705  046426          7$:     MOV     ERWORD,R5       ;GETTING READY TO INDEX
020366  060505                          ADD     R5,R5           ;DOUBLE ERROR WORD
020370  016537  001230  041130          MOV     RHWC-2(R5),REGADR ;FAILING REGISTER ADDRESS
020376  104001                          EMT     1
                                                                ;  COMMAND
020400  000207                          RTS     PC              ;RETURN TO COMPARISON
020402                          8$:                             ;GOOD COMPARISON


                                ;;*****************************************************************
                                ;*TEST 46        UNLOAD COMMAND TEST
                                ;*THE UNLOAD COMMAND WILL BE LOADED INTO RHCS1 WITH GO
                                ;*THEN ALL REGISTERS WILL BE CHECKED
                                ;*RH CLEAR WILL BE GIVEN
                                ;;*****************************************************************
020402  000004                  TST46:  SCOPE
```

```
                                        MOV     #STACK,SP       ;RESET STACK
                                        MOV     #46,TSTNM       ;MOVE #46 TO TEST NUMBER
                                        JSR     PC,CLDISK       ;INIT AND SET UP GENERAL REG.
                                                                ;AND UNIT NUMBER
                                        MOV     #DMD,@RHMR      ;SET DIAGNOSTIC MODE BIT
                                                                ;THIS ENABLES COMMANDS WITHOUT MOL
                                                                ;AND HOLDS RHLA FROM MOVING

                                        MOV     UNLOAD,@RHCS1   ;LOAD UNLOAD COMMAND INTO RH

                                ;SAVE REGISTERS FOR COMPARISON AFTER GO

                                        JSR     R0,SAVER        ;SAVE
                                        RHWC                    ;FROM
                                        REINTO                  ;TO
                                        19.                     ;NUMBER OF REGISTERS SAVED

                                ;GIVE GO TO UNLOAD COMMAND
                                        BIS     #GO,@RHCS1      ;GO TO UNLOAD COMMAND

                                ;CHANGE SAVED REGISTERS TO EXPECTED VALUES
                                        BIS     #GO,REINTO+6    ;SAVED RHCS1
                                        BIS     #PIP,REINTO+30 ;SAVED RHDS1
                                        BIC     #DRY,REINTO+30 ;SAVED RHDS1

                                        TST     $PASS           ;IS THIS FIRST PASS
                                        BNE     5$              ;BRANCH IF NOT FIRST PASS
                                        BIT     #SW13,@SWR      ;INHIBIT ERROR PRINT HIGH?
                                        BNE     5$              ;BRANCH IF SW13 HIGH
                                        TYPE    ,65$            ;;TYPE ASCIZ STRING
                                        BR      64$             ;;GET OVER THE ASCIZ
                                ;;65$:  .ASCIZ  <CRLF>/IF DRIVE CONNECTED "STAND BY" LAMP SHOULD BE LIT ON DRIVE #/
                                64$:
                                        MOV     UNIT,-(SP)      ;UNIT UNDER TEST
                                        TYPDS
                                        TYPE    ,$CRLF          ;CR-LF

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

                                5$:     JSR     R0,SAVER        ;SAVE
                                        RHWC                    ;FROM
                                        WRFROM                  ;TO
                                        19.                     ;NUMBER OF REGISTERS SAVED

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


                                ;COMPARE REGISTERS BEFORE UNLOAD COMMAND
                                ;WITH AFTER GO

                                        JSR     R0,COMPAR       ;COMPARE
                                        REINTO                  ;GOOD BUFFER
```

```
020662  001510                          WRFROM                  ;TEST BUFFER
020664  000023                          19.                     ;NUMBER
020666  020674                          1$                      ;RETURN FOR ERROR
020670  020674                          1$                      ;SAME
020672  020714                          2$                      ;RETURN FOR GOOD COMPARISON
020674  013705  046426          1$:     MOV     ERWORD,R5       ;GETTING READY TO INDEX
020700  060505                          ADD     R5,R5           ;DOUBLE ERROR WORD
020702  016537  001230  041130          MOV     RHWC-2(R5),REGADR ;FAILING REGISTER ADDRESS

020710  104001                          EMT     1
                                                                ;AFTER UNLOAD COMMAND
                                                                ;WITH GO IS GIVEN
020712  000207                          RTS     PC              ;RETURN TO COMPARISON

                                ;NOW GIVE INIT AND GET ALL GO AND PIP DOWN
020714  052712  000040          2$:     BIS     #CLR,@R2        ;RH INITILIZE
020720  013712  001374                  MOV     UNIT,@R2        ;REINSTATE UNIT NUMBER
020724  012777  000001  160326          MOV     #DMD,@RHMR      ;SET DIAGNOSTIC MODE BIT
                                                                ;THIS ENABLES COMMANDS WITHOUT MOL
                                                                ;AND HOLDS RHLA FROM MOVING

                                ;CHANGE REGISTERS TO EXPECTED VALUE
020732  042737  000001  002562          BIC     #GO,REINTO+6    ;SAVED RHCS1
020740  042737  020000  002604          BIC     #PIP,REINTO+30 ;SAVED RHDS1
020746  052737  000200  002604          BIS     #DRY,REINTO+30 ;SAVED RHDS1
020754  017737  160314  002616          MOV     @RHLA,REINTO+42;SAVED RHLA

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

020762  004037  042152                  JSR     R0,SAVER        ;SAVE
020766  001232                          RHWC                    ;FROM
020770  001510                          WRFROM                  ;TO
020772  000023                          19.                     ;NUMBER OF REGISTERS SAVED

                                ;COMPARE REGISTERS AFTER INITIALIZE
                                ;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
020774  113737  002601  001535          MOVB    REINTO+25,WRFROM+25;SAVE UPPER RHAS

021002  004037  042354                  JSR     R0,COMPAR       ;COMPARE
021006  002554                          REINTO                  ;GOOD BUFFER
021010  001510                          WRFROM                  ;TEST BUFFER
021012  000023                          19.                     ;NUMBER OF REGISTERS TO BE
                                                                ;COMPARED
021014  021022                          3$                      ;RETURN POINT FOR ERROR
021016  021022                          3$                      ;SAME
021020  021042                          4$                      ;RETURN POINT FOR GOOD COMPARISON

021022  013705  046426          3$:     MOV     ERWORD,R5       ;GETTING READY TO INDEX
021026  060505                          ADD     R5,R5           ;DOUBLE ERROR WORD
021030  016537  001230  041130          MOV     RHWC-2(R5),REGADR ;FAILING REGISTER ADDRESS
021036  104001                          EMT     1
                                                                ;CONTENTS AFTER GIVING AN
                                                                ;UNLOAD COMMAND
```

```
                                        RTS     PC              ;RETURN TO COMPARISON
                                4$:                             ;GOOD COMPARISON

                                        BIT     #SW4,@SWR       ;TEST FOR NO OFFSET OR RTC
                                        BEQ     6$              ;IF = 0, DO THE NEXT TWO TESTS
                                        JMP     TST51           ;SKIP THE NEXT TWO TESTS
                                6$:                             ;CONTINUE WITH NEXT TWO TESTS

                                ;;*****************************************************************
                                ;*TEST 47       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
                                ;;*****************************************************************
021056  000004                  TST47:  SCOPE
021060  012706  001100                  MOV     #STACK,SP       ;RESET STACK
021064  012737  000047  001432          MOV     #47,TSTNM       ;MOVE #47 TO TEST NUMBER
021072  004737  041460                  JSR     PC,CLDISK       ;INIT AND SET UP GENERAL REG.
                                                                ;AND UNIT NUMBER
021076  012777  000001  160154          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
021104  052777  000004  160146          BIS     #MINX,@RHMR     ;SET INDEX PULSE
021112  042777  000004  160140          BIC     #MINX,@RHMR     ;CLEAR INDEX

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

021120  017777  160152  160124          MOV     @RHCC,@RHCA     ;SET DESIRED CYLINDER TO RHCC
021126  013711  001472                  MOV     SEECOM,@R1      ;SEEK COMMAND TO RHCS1
021132  005211                          INC     @R1             ;GO TO SEEK COMMAND

                                ;FOUR SECTOR CLOCKS ARE GIVEN TO TAKE POSITIONER OFF OFFSET POSITION
021134  012700  000004                  MOV     #4,R0           ;COUNTER
021140  012777  000011  160112  5$:     MOV     #MSTCK!DMD,@RHMR        ;SET SECTOR CLOCK
021146  012777  000001  160104          MOV     #DMD,@RHMR      ;RESET SECTOR CLOCK
021154  005300                          DEC     R0              ;COUNT
021156  001370                          BNE     5$              ;BRANCH IF NOT COMPLETE

021160  004737  041460                  JSR     PC,CLDISK       ;INIT AND SET UP GENERAL REG.
                                                                ;AND UNIT NUMBER
021164  012777  000001  160066          MOV     #DMD,@RHMR      ;SET DIAGNOSTIC MODE BIT
                                                                ;THIS ENABLES COMMANDS WITHOUT MOL
                                                                ;AND HOLDS RHLA FROM MOVING

021172  013777  001474  160040          MOV     OFSETC,@RHCS1   ;LOAD AN OFFSET BIT
021200  012777  000001  160042          MOV     #OF25,@RHOF     ;SET AN OFFSET BIT

                                ;SAVE REGISTERS FOR COMPARISON AFTER GO
021206  004037  042152                  JSR     R0,SAVER        ;SAVE
021212  001232                          RHWC                    ;FROM
021214  002554                          REINTO                  ;TO
021216  000023                          19.                     ;NUMBER OF REGISTERS SAVED
```

```
                                        ;GIVE GO TO OFFSET COMMAND
021220  052777  000001  160012            BIS     #GO,@RHCS1      ;GO TO OFFSET COMMAND

                                        ;CHANGE SAVED REGISTERS TO EXPECTED VALUES
021226  052737  000001  002562            BIS     #GO,REINTO+6    ;SAVED RHCS1
021234  052737  020000  002604            BIS     #PIP,REINTO+30 ;SAVED RHDS1
021242  042737  000200  002604            BIC     #DRY,REINTO+30 ;SAVED RHDS1

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

021250  004037  042152                    JSR     R0,SAVER        ;SAVE
021254  001232                            RHWC                    ;FROM
021256  001510                            WRFROM                  ;TO
021260  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
021262  113737  002601  001535            MOVB    REINTO+25,WRFROM+25;SAVE UPPER RHAS


                                        ;COMPARE REGISTERS BEFORE OFFSET COMMAND
                                        ;WITH AFTER GO

021270  004037  042354                    JSR     R0,COMPAR       ;COMPARE
021274  002554                            REINTO                  ;GOOD BUFFER
021276  001510                            WRFROM                  ;TEST BUFFER
021300  000023                            19.                     ;NUMBER
021302  021310                            1$                      ;RETURN FOR ERROR
021304  021310                            1$                      ;SAME
021306  021330                            2$                      ;RETURN FOR GOOD COMPARISON

021310  013705  046426          1$:       MOV     ERWORD,R5       ;GETTING READY TO INDEX
021314  060505                            ADD     R5,R5           ;DOUBLE ERROR WORD
021316  016537  001230  041130            MOV     RHWC-2(R5),REGADR ;FAILING REGISTER ADDRESS
021324  104001                            EMT     1
                                                                  ;AFTER OFFSET COMMAND
                                                                  ;WITH GO IS GIVEN
021326  000207                            RTS     PC              ;RETURN TO COMPARISON

                                        ;NOW GIVE INIT AND GET ALL GO AND PIP DOWN

021330  052712  000040          2$:       BIS     #CLR,@R2        ;RH INITILIZE
021334  013712  001374                    MOV     UNIT,@R2        ;REINSTATE UNIT NUMBER
021340  012777  000001  157712            MOV     #DMD,@RHMR      ;SET DIAGNOSTIC MODE BIT
                                                                  ;THIS ENABLES COMMANDS WITHOUT MOL
                                                                  ;AND HOLDS RHLA FROM MOVING

                                        ;CHANGE REGISTERS TO EXPECTED VALUE
021346  042737  000001  002562            BIC     #GO,REINTO+6    ;SAVED RHCS1
021354  042737  000001  002572            BIC     #OF25,REINTO+16;SAVED RHOF
021362  042737  020000  002604            BIC     #PIP,REINTO+30 ;SAVED RHDS1
021370  052737  000200  002604            BIS     #DRY,REINTO+30 ;SAVED RHDS1
021376  017737  157672  002616            MOV     @RHLA,REINTO+42;SAVED RHLA
```

```
                                        ;AFTER INITIALIZE SAVE REGISTERS SO THAT
                                        ;COMPARES CAN BE DONE
021404  004037  042152                        JSR     R0,SAVER        ;SAVE
021410  001232                                RHWC                    ;FROM
021412  001510                                WRFROM                  ;TO
021414  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
021416  113737  002601  001535                MOVB    REINTO+25,WRFROM+25;SAVE UPPER RHAS

                                        ;COMPARE REGISTERS AFTER INITIALIZE
021424  004037  042354                        JSR     R0,COMPAR       ;COMPARE
021430  002554                                REINTO                  ;GOOD BUFFER
021432  001510                                WRFROM                  ;TEST BUFFER
021434  000023                                19.                     ;NUMBER OF REGISTERS TO BE
                                                                        ;COMPARED
021436  021444                                3$                      ;RETURN POINT FOR ERROR
021440  021444                                3$                      ;SAME
021442  021464                                4$                      ;RETURN POINT FOR GOOD COMPARISON

021444  013705  046426        3$:     MOV     ERWORD,R5       ;GETTING READY TO INDEX
021450  060505                        ADD     R5,R5           ;DOUBLE ERROR WORD
021452  016537  001230  041130        MOV     RHWC-2(R5),REGADR ;FAILING REGISTER ADDRESS
021460  104001                        EMT     1
                                                                ;CONTENTS AFTER GIVING AN
                                                                ;INITIALIZE FOLLOWING A
                                                                ;OFFSET COMMAND
021462  000207                        RTS     PC              ;RETURN TO COMPARISON

021464                          4$:                             ;GOOD COMPARISON

                                        ;;**********************************************************************
                                        ;*TEST 50       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
                                        ;;**********************************************************************
021464  000004                    TST50:  SCOPE
021466  012706  001100                    MOV     #STACK,SP       ;RESET STACK
021472  012737  000050  001432            MOV     #50,TSTNM       ;MOVE #50 TO TEST NUMBER
021500  004737  041460                    JSR     PC,CLDISK       ;INIT AND SET UP GENERAL REG.
                                                                ;AND UNIT NUMBER
021504  012777  000001  157546            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
021512  052777  000004  157540            BIS     #MINX,@RHMR     ;SET INDEX PULSE
021520  042777  000004  157532            BIC     #MINX,@RHMR     ;CLEAR INDEX


021526  013777  001476  157504            MOV     RETCL,@RHCS1    ;LOAD RETURN TO CENTER LINE COMMAND INTO RHCS1
```

```
                                        ;SAVE REGISTERS FOR COMPARISON AFTER GO
021534  004037  042152                          JSR     R0,SAVER        ;SAVE
021540  001232                                  RHWC                    ;FROM
021542  002554                                  REINTO                  ;TO
021544  000023                                  19.                     ;NUMBER OF REGISTERS SAVED

                                        ;GIVE GO TO RETURN TO CENTER LINE COMMAND
021546  052777  000001  157464                  BIS     #GO,@RHCS1      ;GO TO RETURN TO CENTER COMMAND


                                        ;FOUR SECTOR CLOCKS ARE GIVEN TO TAKE POSITIONER TO CENTER LINE
021554  012700  000004                          MOV     #4,R0           ;COUNTER
021560  012777  000011  157472  5$:     MOV     #MSTCK!DMD,@RHMR;SET SECTOR CLOCK
021566  012777  000001  157464          MOV     #DMD,@RHMR      ;RESET SECTOR CLOCK
021574  005300                          DEC     R0              ;COUNT
021576  001370                          BNE     5$              ;BRANCH IF NOT COMPLETE


                                        ;CHANGE SAVED REGISTERS TO EXPECTED VALUES
021600  052737  000001  002562          BIS     #GO,REINTO+6    ;SAVED RHCS1
021606  052737  020000  002604          BIS     #PIP,REINTO+30 ;SAVED RHDS1
021614  042737  000200  002604          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
021622  004037  042152                          JSR     R0,SAVER        ;SAVE
021626  001232                                  RHWC                    ;FROM
021630  001510                                  WRFROM                  ;TO
021632  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
021634  113737  002601  001535                  MOVB    REINTO+25,WRFROM+25;SAVE UPPER RHAS


                                        ;COMPARE REGISTERS BEFORE RETURN TO CENTER LINE COMMAND
                                        ;WITH AFTER GO
021642  004037  042354                          JSR     R0,COMPAR       ;COMPARE
021646  002554                                  REINTO                  ;GOOD BUFFER
021650  001510                                  WRFROM                  ;TEST BUFFER
021652  000023                                  19.                     ;NUMBER
021654  021662                                  1$                      ;RETURN FOR ERROR
021656  021662                                  1$                      ;SAME
021660  021702                                  2$                      ;RETURN FOR GOOD COMPARISON

021662  013705  046426          1$:     MOV     ERWORD,R5       ;GETTING READY TO INDEX
021666  060505                          ADD     R5,R5           ;DOUBLE ERROR WORD
021670  016537  001230  041130          MOV     RHWC-2(R5),REGADR ;FAILING REGISTER ADDRESS
021676  104001                          EMT     1
                                                                ;AFTER RETURN TO CENTER LINE COMMAND
                                                                ;WITH GO IS GIVEN
021700  000207                          RTS     PC              ;RETURN TO COMPARISON

                                        ;NOW GIVE INIT AND GET ALL GO AND PIP DOWN
```

```
1091 021702 052712 000040          2$:    BIS    #CLR,@R2          ;RH INITILIZE
1092 021706 013712 001374                 MOV    UNIT,@R2          ;REINSTATE UNIT NUMBER
1093 021712 012777 000001 157340          MOV    #DMD,@RHMR        ;SET DIAGNOSTIC MODE BIT
1094                                                                ;THIS ENABLES COMMANDS WITHOUT MOL
1095                                                                ;AND HOLDS RHLA FROM MOVING
1096
1097                               ;CHANGE REGISTERS TO EXPECTED VALUE
1098 021720 042737 000001 002562          BIC    #GO,REINTO+6      ;SAVED RHCS1
1099 021726 042737 020000 002604          BIC    #PIP,REINTO+30    ;SAVED RHDS1
1100 021734 052737 000200 002604          BIS    #DRY,REINTO+30    ;SAVED RHDS1
1101 021742 017737 157326 002616          MOV    @RHLA,REINTO+42   ;SAVED RHLA
1102
1103                               ;AFTER INITIALIZE SAVE REGISTERS SO THAT
1104                               ;COMPARES CAN BE DONE
1105 021750 004037 042152                 JSR    R0,SAVER          ;SAVE
1106 021754 001232                        RHWC                     ;FROM
1107 021756 001510                        WRFROM                   ;TO
1108 021760 000023                        19.                      ;NUMBER OF REGISTERS SAVED
1109
1110                               ;AS UPPER BYTE OF RHAS CAN BE CHANGING IN A DUAL PORT
1111                               ;OPERATION THE UPPER BYTE OF RHAS WILL BE SAVED AS IS
1112                               ;SO THAT THE COMPARES ARE ONLY VALID FOR THE LOWER BYTE
1113 021762 113737 002601 001535          MOVB   REINTO+25,WRFROM+25;SAVE UPPER RHAS
1114
1115
1116                               ;COMPARE REGISTERS AFTER INITIALIZE
1117 021770 004037 042354                 JSR    R0,COMPAR         ;COMPARE
1118 021774 002554                        REINTO                   ;GOOD BUFFER
1119 021776 001510                        WRFROM                   ;TEST BUFFER
1120 022000 000023                        19.                      ;NUMBER OF REGISTERS TO BE
1121                                                                ;COMPARED
1122 022002 022010                        3$                       ;RETURN POINT FOR ERROR
1123 022004 022010                        3$                       ;SAME
1124 022006 022030                        4$                       ;RETURN POINT FOR GOOD COMPARISON
1125
1126
1127 022010 013705 046426          3$:    MOV    ERWORD,R5         ;GETTING READY TO INDEX
1128 022014 060505                        ADD    R5,R5             ;DOUBLE ERROR WORD
1129 022016 016537 001230 041130          MOV    RHWC-2(R5),REGADR ;FAILING REGISTER ADDRESS
1130 022024 104001                        EMT    1
1131                                                                ;CONTENTS AFTER GIVING AN
1132                                                                ;INITIALIZE FOLLOWING A RETURN TO
1133                                                                ;CENTER LINE COMMAND
1134 022026 000207                        RTS    PC                ;RETURN TO COMPARISON
1135
1136 022030                        4$:                              ;GOOD COMPARISON
1137
1144                               ;;*****************************************************************
                                   ;*TEST 51       RECALIBRATE COMMAND TEST
                                   ;*THE RECALIBRATE COMMAND WILL BE LOADED INTO RHCS1 WITH GO
                                   ;*THEN ALL REGISTERS WILL BE CHECKED
                                   ;*RH CLEAR WILL BE GIVEN
                                   ;*THEN ALL REGISTERS WILL BE CHECKED
                                   ;;*****************************************************************
 022030 000004                     TST51: SCOPE
1145 022032 012706 001100                 MOV    #STACK,SP         ;RESET STACK
1146 022036 012737 000051 001432          MOV    #51,TSTNM         ;MOVE #51 TO TEST NUMBER
```

```
1147 022044  004737  041460                JSR     PC,CLDISK       ;INIT AND SET UP GENERAL REG.
1148                                                                 ;AND UNIT NUMBER
1149 022050  012777  000001  157202        MOV     #DMD,@RHMR      ;SET DIAGNOSTIC MODE BIT
1150                                                                 ;THIS ENABLES COMMANDS WITHOUT MOL
1151                                                                 ;AND HOLDS RHLA FROM MOVING
1152
1153
1154                              ;GIVE ONE INDEX PULSE TO CLEAR RHLA BEFORE THE START OF THIS TEST
1155
1156 022056  052777  000004  157174        BIS     #MINX,@RHMR     ;SET INDEX PULSE
1157 022064  042777  000004  157166        BIC     #MINX,@RHMR     ;CLEAR INDEX
1158
1159 022072  013777  001446  157140        MOV     RECALI,@RHCS1   ;LOAD RECALIBRATE COMMAND INTO RHCS1
1160
1161                              ;SAVE REGISTERS FOR COMPARISON AFTER GO
1162 022100  004037  042152                JSR     R0,SAVER        ;SAVE
1163 022104  001232                        RHWC                    ;FROM
1164 022106  002554                        REINTO                  ;TO
1165 022110  000023                        19.                     ;NUMBER OF REGISTERS SAVED
1166
1167                              ;GIVE GO TO RECALIBRATE COMMAND
1168 022112  052777  000001  157120        BIS     #GO,@RHCS1      ;GO TO RECALIBRATE COMMAND
1169
1170
1171                              ;FOUR SECTOR CLOCKS ARE GIVEN TO TAKE POSITIONER TO CYLINDER 0
1172 022120  012700  000004                MOV     #4,R0           ;COUNTER
1173 022124  012777  000011  157126  5$:   MOV     #MSTCK!DMD,@RHMR;SET SECTOR CLOCK
1174 022132  012777  000001  157120        MOV     #DMD,@RHMR      ;RESET SECTOR CLOCK
1175 022140  005300                        DEC     R0              ;COUNT
1176 022142  001370                        BNE     5$              ;BRANCH IF NOT COMPLETE
1177
1178
1179                              ;CHANGE SAVED REGISTERS TO EXPECTED VALUES
1180 022144  052737  000001  002562        BIS     #GO,REINTO+6    ;SAVED RHCS1
1181 022152  052737  020000  002604        BIS     #PIP,REINTO+30 ;SAVED RHDS1
1182 022160  042737  000200  002604        BIC     #DRY,REINTO+30 ;SAVED RHDS1
1183
1184                              ;AFTER GO HAS BEEN GIVEN TO RECALIBRATE COMMAND
1185                              ;SAVE REGISTERS AGAIN SO THAT COMPARISONS CAN
1186                              ;BE DONE
1187 022166  004037  042152                JSR     R0,SAVER        ;SAVE
1188 022172  001232                        RHWC                    ;FROM
1189 022174  001510                        WRFROM                  ;TO
1190 022176  000023                        19.                     ;NUMBER OF REGISTERS SAVED
1191
1192                              ;AS UPPER BYTE OF RHAS CAN BE CHANGING IN A DUAL PORT
1193                              ;OPERATION THE UPPER BYTE OF RHAS WILL BE SAVED AS IS
1194                              ;SO THAT THE COMPARES ARE ONLY VALID FOR THE LOWER BYTE
1195 022200  113737  002601  001535        MOVB    REINTO+25,WRFROM+25;SAVE UPPER RHAS
1196
1197
1198                              ;COMPARE REGISTERS BEFORE RECALIBRATE COMMAND
1199                              ;WITH AFTER GO
1200 022206  004037  042354                JSR     R0,COMPAR       ;COMPARE
1201 022212  002554                        REINTO                  ;GOOD BUFFER
1202 022214  001510                        WRFROM                  ;TEST BUFFER
1203 022216  000023                        19.                     ;NUMBER
```

```
1204 022220  022226                        1$                      ;RETURN FOR ERROR
1205 022222  022226                        1$                      ;SAME
1206 022224  022246                        2$                      ;RETURN FOR GOOD COMPARISON
1207
1208 022226  013705  046426          1$:   MOV     ERWORD,R5         ;GETTING READY TO INDEX
1209 022232  060505                  ADD     R5,R5                   ;DOUBLE ERROR WORD
1210 022234  016537  001230  041130        MOV     RHWC-2(R5),REGADR ;FAILING REGISTER ADDRESS
1211 022242  104001                  EMT     1
1212                                                                 ;AFTER RECALIBRATE COMMAND
1213                                                                 ;WITH GO IS GIVEN
1214 022244  000207                  RTS     PC                      ;RETURN TO COMPARISON
1215
1216                              ;NOW GIVE INIT AND GET ALL GO AND PIP DOWN
1217
1218 022246  052712  000040          2$:   BIS     #CLR,@R2          ;RH INITILIZE
1219 022252  013712  001374          MOV     UNIT,@R2                ;REINSTATE UNIT NUMBER
1220 022256  012777  000001  156774        MOV     #DMD,@RHMR        ;SET DIAGNOSTIC MODE BIT
1221                                                                 ;THIS ENABLES COMMANDS WITHOUT MOL
1222                                                                 ;AND HOLDS RHLA FROM MOVING
1223
1224                              ;CHANGE REGISTERS TO EXPECTED VALUE
1225 022264  042737  000001  002562        BIC     #GO,REINTO+6    ;SAVED RHCS1
1226 022272  042737  020000  002604        BIC     #PIP,REINTO+30 ;SAVED RHDS1
1227 022300  052737  000200  002604        BIS     #DRY,REINTO+30 ;SAVED RHDS1
1228 022306  017737  156762  002616        MOV     @RHLA,REINTO+42;SAVED RHLA
1229
1230                              ;AFTER INITIALIZE SAVE REGISTERS SO THAT
1231                              ;COMPARES CAN BE DONE
1232 022314  004037  042152                JSR     R0,SAVER        ;SAVE
1233 022320  001232                  RHWC                            ;FROM
1234 022322  001510                  WRFROM                          ;TO
1235 022324  000023                  19.                             ;NUMBER OF REGISTERS SAVED
1236
1237                              ;AS UPPER BYTE OF RHAS CAN BE CHANGING IN A DUAL PORT
1238                              ;OPERATION THE UPPER BYTE OF RHAS WILL BE SAVED AS IS
1239                              ;SO THAT THE COMPARES ARE ONLY VALID FOR THE LOWER BYTE
1240 022326  113737  002601  001535        MOVB    REINTO+25,WRFROM+25;SAVE UPPER RHAS
1241
1242
1243                              ;COMPARE REGISTERS AFTER INITIALIZE
1244 022334  004037  042354                JSR     R0,COMPAR       ;COMPARE
1245 022340  002554                  REINTO                          ;GOOD BUFFER
1246 022342  001510                  WRFROM                          ;TEST BUFFER
1247 022344  000023                  19.                             ;NUMBER OF REGISTERS TO BE
1248                                                                 ;COMPARED
1249 022346  022354                  3$                              ;RETURN POINT FOR ERROR
1250 022350  022354                  3$                              ;SAME
1251 022352  022374                  4$                              ;RETURN POINT FOR GOOD COMPARISON
1252
1253 022354  013705  046426          3$:   MOV     ERWORD,R5         ;GETTING READY TO INDEX
1254 022360  060505                  ADD     R5,R5                   ;DOUBLE ERROR WORD
1255 022362  016537  001230  041130        MOV     RHWC-2(R5),REGADR ;FAILING REGISTER ADDRESS
1256 022370  104001                  EMT     1
1257                                                                 ;CONTENTS AFTER GIVING AN
1258                                                                 ;INITIALIZE FOLLOWING A
1259                                                                 ;RECALIBRATE COMMAND
1260 022372  000207                  RTS     PC                      ;RETURN TO COMPARISON
```

```
1261
1262 022374                        4$:                             ;GOOD COMPARISON
1263
1270                               ;;*****************************************************************
                                   ;*TEST 52       RELEASE COMMAND TEST
                                   ;*THE RELEASE COMMAND WILL BE LOADED INTO RHCS1 WITH GO
                                   ;*THEN ALL REGISTERS WILL BE CHECKED
                                   ;*RH CLEAR WILL BE GIVEN
                                   ;*THEN ALL REGISTERS WILL BE CHECKED
                                   ;;*****************************************************************
    022374  000004                 TST52:  SCOPE
1271 022376  012706  001100                MOV     #STACK,SP       ;RESET STACK
1272 022402  012737  000052  001432        MOV     #52,TSTNM       ;MOVE #52 TO TEST NUMBER
1273 022410  004737  041460                JSR     PC,CLDISK       ;INIT AND SET UP GENERAL REG.
1274                                                               ;AND UNIT NUMBER
1275 022414  012777  000001  156636        MOV     #DMD,@RHMR      ;SET DIAGNOSTIC MODE BIT
1276                                                               ;THIS ENABLES COMMANDS WITHOUT MOL
1277                                                               ;AND HOLDS RHLA FROM MOVING
1278
1279 022422  013777  001452  156610        MOV     RELEASE,@RHCS1 ;LOAD RELEASE COMMAND INTO RHCS1
1280
1281                               ;SAVE REGISTERS FOR COMPARISON AFTER GO
1282 022430  004037  042152                JSR     R0,SAVER        ;SAVE
1283 022434  001232                        RHWC                    ;FROM
1284 022436  002554                        REINTO                  ;TO
1285 022440  000023                        19.                     ;NUMBER OF REGISTERS SAVED
1286
1287                               ;GIVE GO TO RELEASE COMMAND
1288 022442  052777  000001  156570        BIS     #GO,@RHCS1      ;GO TO RELEASE COMMAND
1289 022450  052777  000001  156602        BIS     #DMD,@RHMR      ;SET DMD TO HOLD RHLA
1290
1291                               ;CHANGE SAVED REGISTERS TO EXPECTED VALUES
1292                               ;AFTER GO HAS BEEN GIVEN TO RELEASE COMMAND
1293 022456  052737  000001  002602        BIS     #DMD,REINTO+26;SAVED RHMR
1294 022464  017737  156604  002616        MOV     @RHLA,REINTO+42;SAVED RHLA
1295
1296
1297                               ;SAVE REGISTERS AGAIN SO THAT COMPARISONS CAN
1298                               ;BE DONE
1299 022472  004037  042152                JSR     R0,SAVER        ;SAVE
1300 022476  001232                        RHWC                    ;FROM
1301 022500  001510                        WRFROM                  ;TO
1302 022502  000023                        19.                     ;NUMBER OF REGISTERS SAVED
1303
1304                               ;AS UPPER BYTE OF RHAS CAN BE CHANGING IN A DUAL PORT
1305                               ;OPERATION THE UPPER BYTE OF RHAS WILL BE SAVED AS IS
1306                               ;SO THAT THE COMPARES ARE ONLY VALID FOR THE LOWER BYTE
1307 022504  113737  002601  001535        MOVB    REINTO+25,WRFROM+25;SAVE UPPER RHAS
1308
1309
1310                               ;COMPARE REGISTERS BEFORE RELEASE COMMAND
1311                               ;WITH AFTER GO
1312 022512  004037  042354                JSR     R0,COMPAR       ;COMPARE
1313 022516  002554                        REINTO                  ;GOOD BUFFER
1314 022520  001510                        WRFROM                  ;TEST BUFFER
1315 022522  000023                        19.                     ;NUMBER
1316 022524  022532                        1$                      ;RETURN FOR ERROR
```

```
1317 022526  022532                        1$                      ;SAME
1318 022530  022552                        2$                      ;RETURN FOR GOOD COMPARISON
1319
1320 022532  013705  046426          1$:   MOV     ERWORD,R5         ;GETTING READY TO INDEX
1321 022536  060505                        ADD     R5,R5             ;DOUBLE ERROR WORD
1322 022540  016537  001230  041130        MOV     RHWC-2(R5),REGADR ;FAILING REGISTER ADDRESS
1323 022546  104001                        EMT     1
1324                                                                 ;AFTER RELEASE COMMAND
1325                                                                 ;WITH GO IS GIVEN
1326 022550  000207                        RTS     PC                ;RETURN TO COMPARISON
1327 022552                          2$:
1328                                 ;;*****************************************************************
                                     ;*TEST 53        MAKE CURRENT CYLINDER = 0
                                     ;;*****************************************************************
     022552  000004                 TST53:  SCOPE
     022554  012706  001100                MOV     #STACK,SP         ;RESET STACK
     022560  012737  000053  001432        MOV     #53,TSTNM         ;MOVE #53 TO TEST NUMBER
     022566  004737  041460                JSR     PC,CLDISK         ;INIT DRIVE
     022572  012777  000001  156460        MOV     #DMD,@RHMR        ;SET DIAGNOSTIC MODE
     022600  004037  044032                JSR     R0,MAKECYL        ;DO SEEK COMMAND FOLLOWED BY AN INIT TO
     022604  000000                        .WORD   0                 ;CHANGE RHCC TO CYLINDER 0
1329
1338                                 ;;*****************************************************************
                                     ;*TEST 54        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
                                     ;;*****************************************************************
     022606  000004                 TST54:  SCOPE
1339 022610  012706  001100                MOV     #STACK,SP         ;RESET STACK
1340 022614  012737  000054  001432        MOV     #54,TSTNM         ;MOVE #54 TO TEST NUMBER
1341 022622  004737  041460                JSR     PC,CLDISK         ;INIT AND SET UP GENERAL REGISTERS
1342
1343                                 ;THESE ARE REGULAR SET UPS FOR SEARCH COMMAND
1344 022626  012777  000025  156410        MOV     #21.,@RHDST       ;DESIRED SECTOR/TRACK REGISTER
1345                                                                 ;TRACK 0 SECTOR 21
1346 022634  005077  156412                CLR     @RHCA             ;DESIRED CYLINDER =0
1347 022640  012777  010000  156402        MOV     #FMT22,@RHOF      ;FORMAT BIT=1 (16 BITS PER WORD)
1348 022646  013711  001454                MOV     SERCH,@R1         ;FILL SEARCH COMMAND IN RHCS1
1349
1350                                 ;NOW SAVE REGISTERS STARTING FROM RHWC IN WRITE FROM BUFFER
1351 022652  004037  042152                JSR     R0,SAVER          ;SAVE REGISTERS FOR COMPARISON
1352                                                                 ;AT THE END OF THE SEARCH
1353 022656  001232                        RHWC                      ;START SAVING FROM RHWC
1354 022660  002554                        REINTO                    ;SAVE INTO REINTO
1355 022662  000023                        19.                       ;NUMBER OF REGISTERS SAVED
1356
1357 022664  004737  041514                JSR     PC,CHECKT         ;CHECK THAT DVA,RDY,DPR,DRY = 1
     022670  104401  062726                TYPE    ,CPHALT           ;AND THAT NO OTHERS = 1.  CANNOT CON-
                                                                     ;TINUE TESTING IF BOTH AREN'T TRUE
     022674  000000                        HALT                      ;STOP THE TEST
1358
1359                                 ;NOW THE DIAGNOSTIC MODE BIT WILL BE SET
1360                                 ;AND THE SEARCH OPERATION STARTED
```

```
                                                ;THIS WILL HAVE THE EXPECTED
                                                ;VALUE OF RHLA REGISTER


                        MOV     RHMR,R0 ;NOW R0 HAS MAINTENANCE REG. ADDR.
                        MOV     @RHDST,R3       ;GET DESIRED SECTOR/TRACK REG.
                        BIC     #177400,R3      ;GET SECTOR ONLY
                        MOV     R3,SECTR        ;DUPLICATE SECTOR
                        MOV     #DMD,@R0        ;S
156304                  BIS     #GO,@RHCS1      ;GO
                        BIS     #MSTCK,@R0      ;SET SECTOR CLOCK
                        BIC     #MSTCK,@R0      ;CLEAR SECTOR CLOCK
                        NOP                     ;ALLOW TIME BETWEEN SECTOR CLOCKS
                        BIS     #MSTCK,@R0      ;SET SECTOR CLOCK
                        BIC     #MSTCK,@R0      ;CLEAR SECTOR CLOCK
                        NOP                     ;ALLOW TIME BETWEEN SECTOR CLOCKS
                        BIS     #MINX!MSTCK,@R0 ;SET INDEX AND SECTOR CLOCK
                        MOV     #DMD,@R0        ;RESET INDEX AND SECTOR CLOCK
                        TST     R3              ;IF SECTOR REQUIRED JUMP OUT
                        BEQ     11$             ;BRANCH OF SECTOR ZERO REQUIRED

                ;AFTER THE INDEX PULSE RHLA WILL BE CHECKED TO BE ZERO
                ;AND $TMP4 WILL BE SET UP TO COUNT BYTES
001140  001206  1$:     MOV     #608.,$TMP4     ;THERE ARE 608 BYTES PER SECTOR
156266  001126          MOV     @RHLA,$BDDAT    ;SAVE RHLA
156230  001320          MOV     @RHDST,DST      ;SAVE DESIRED SECTOR TRACK
001200  001126          CMP     $TMP1,$BDDAT ;RHLA SHOULD BE HAVE EXTENSION
                                                ;FIELD EQUAL TO ZERO
001414                  BEQ     2$              ;BRANCH IF GOOD
052352  001202          MOV     SECTR,$TMP2     ;GET SECTOR SOUGHT
                        SUB     R3,$TMP2        ;$TMP2 NOW HAS PRESENT SECTOR
001140                  MOV     #608.,-(SP)     ;NUMBER OF BYTES PER SECTOR
001206                  SUB     $TMP4,(SP)      ;(SP)HAS PRESENT BYTE NUMBER
001204                  MOV     (SP)+,$TMP3     ;PRESENT BYTE NUMBER
104024                  EMT     24
                                                ;SECTOR IS IN ERROR


                ;NOW THE 304 WORDS WILL START
                ;FOR FIRST BYTE CLOCK WILL BE INDEPENDENT OF
                ;SECTOR CLOCK THEN IT WILL COINCIDE FOREVER TILL
                ;THE BEGINNING OF NEXT SECTOR

                ;ONE WORD ONLY THAT IS TWO BYTES

000010          2$:     MOV     #8.,R2          ;BYTE
000002                  MOV     #2,R5           ;BYTES PER WORD
000404                  BR      4$
000012          3$:     BIS     #MSTCK!MCLK,@R0 ;SET SECTOR AND CLOCK
000012                  BIC     #MSTCK!MCLK,@R0 ;CLEAR SECTOR AND CLOCK
000002          4$:     BIS     #MCLK,@R0       ;SET CLOCK
000002                  BIC     #MCLK,@R0       ;CLEAR CLOCK
005302                  DEC     R2              ;BYTE COUNTER
001372                  BNE     4$              ;BRANCH IF BYTE NOT COMPLETE
001206                  DEC     $TMP4           ;BYTE COUNT DOWN
000007                  MOV     #7,R2           ;SETUP FOR SECOND BYTE
005305                  DEC     R5              ;IS WORD COMPLETE?
```

```
1418 023126  001360                        BNE     3$              ;BRENCH IF NOT COMPLETE
1419                                                               ;TO GIVE SECTOR CLOCK AND CLOCK
1420
1421                                ;NOW 303 WORDS ARE LEFT ALL ARE IDENTICAL
1422                                ;THAT IS 606 IDENTICAL BYTES WILL BE GIVEN
1423                                ;RHLA WILL BE CHECKED STAR TO COUNT AFTER
1424                                ;BEGINNING OF SECTOR PULSE
1425                                ;AFTER 128 BYTES (2 BYTES ARE ALREADY GIVEN)
1426                                ;SO 127 MORE
1427                                ;THEN RHLA WILL BE CHECKED AFTER 128 MORE BYTES
1428                                ;THEN RHLA WILL BE CHECKED AFTER 256 MORE BYTES
1429                                ;THEN THE TOTAL OF 608 BYTES WILL BE COMPLETED
1430                                ;AND RHLA WILL BE MADE READY FOR NEXT SECTOR
1431                                ;AND RHLA WILL BE CHECKED
1432
1433 023130  012705  000100                MOV     #64.,R5         ;R5 WILL KEEP TRACK WHEN
1434                                                               ;EXTENSION FIELD IS TO BE CHECKED
1435 023134  012701  000177                MOV     #127.,R1        ;FIRST TIME CHECK EXTENSION FIELD
1436                                                               ;AFTER 127 MORE BYTES
1437 023140  012702  000007        5$:     MOV     #7,R2           ;CLOCKS PER BYTE COUNTER
1438 023144  052710  000012                BIS     #MSTCK!MCLK,@R0 ;SET SECTOR CLOCK AND CLOCK
1439 023150  042710  000012                BIC     #MSTCK!MCLK,@R0 ;CLEAR SECTOR CLOCK AND CLOCK
1440 023154  052710  000002        6$:     BIS     #MCLK,@R0       ;SET CLOCK
1441 023160  042710  000002                BIC     #MCLK,@R0       ;RESET CLOCK
1442 023164  005302                        DEC     R2              ;COUNT DOWN CLOCKS PER BYTE
1443 023166  001372                        BNE     6$              ;BRANCH IF BYTE NOT COMPLETE
1444 023170  005337  001206                DEC     $TMP4           ;COUNT DOWN BYTES
1445 023174  001436                        BEQ     10$             ;BRANCHOUT IF 608 BYTES DONE
1446 023176  005301                        DEC     R1              ;COUNT DOWN NUMBER OF BYTES
1447                                                               ;TO CHECK EXTENSION FIELD
1448 023200  001357                        BNE     5$              ;BRANCH IF EXTENSION FIELD NOT
1449                                                               ;TO BE CHECKED YET
1450
1451                                ;NOW THE EXTENSION FIELD OF THE LOOK AHEAD REGISTER
1452                                ;WILL BE CHECKED
1453
1454 023202  062737  000020  001200        ADD     #20,$TMP1       ;GET TO THE NEXT EXTENSION
1455 023210  017737  156060  001126        MOV     @RHLA,$BDDAT    ;GET RHLA FOR COMPARISON
1456
1457 023216  017737  156022  001320        MOV     @RHDST,DST      ;SAVE DESIRED SECTOR TRACK
1458 023224  023737  001200  001126        CMP     $TMP1,$BDDAT ;CHECK VALUE OF RHLA
1459 023232  001414                        BEQ     7$              ;BRANCH IF GOOD
1460 023234  013737  052352  001202        MOV     SECTR,$TMP2 ;GET SECTOR SOUGHT
1461 023242  160337  001202                SUB     R3,$TMP2        ;$TMP2 NOW HAS PRESENT SECTOR
1462 023246  012746  001140                MOV     #608.,-(SP)     ;NUMBER OF BYTES PER SECTOR
1463 023252  163716  001206                SUB     $TMP4,(SP)      ;(SP) HAS PRESENT BYTE NUMBER
1464 023256  012637  001204                MOV     (SP)+,$TMP3     ;PRESENT BYTE NUMBER
1465 023262  104025                        EMT     25
1466                                                               ;OF A SECTOR IS IN ERROR
1467
1468 023264  060505                7$:     ADD     R5,R5           ;GET NEXT STEP TO CHECK EXTENSION FIELD
1469 023266  010501                        MOV     R5,R1           ;PUT IN COUNTER
1470 023270  000723                        BR      5$              ;BRANCH BACK SECTOR
1471                                                               ;IS NOT COMPLETE
1472 023272  062737  000020  001200 10$:     ADD     #20,$TMP1
1473 023300  052710  000010                BIS     #MSTCK,@R0      ;THESE TWO INSTRUCTIONS GIVE
1474 023304  042710  000010                BIC     #MSTCK,@R0      ;ONE SECTOR CLOCK EXTRA
```

```
1475 023310  000240                        NOP                     ;ALLOW TIME BETWEEN SECTOR CLOCK
1476 023312  052710  000010                BIS     #MSTCK,@R0      ;THESE TWO INSTRUCTIONS GIVE
1477 023316  042710  000010                BIC     #MSTCK,@R0      ;ONE SECTOR CLOCK EARLY
1478                                                                ;BEFORE THE NEXT SECTOR
1479 023322  005303                        DEC     R3              ;IS REQUIRED NO OF SECTORS COMPLETE
1480 023324  001223                        BNE     1$              ;BRANCH IF NOT
1481
1482                                ;NOW THE REQUIRED SECTOR IS REACHED
1483                                ;ONE SECTOR CLOCK WILL BE GIVEN TO GET SECTOR PULSE
1484                                ;DOWN AND HENCE ATA UP
1485
1486 023326  012702  000010          11$:    MOV     #8.,R2          ;8 CLOCKS
1487 023332  052710  000002          12$:    BIS     #MCLK,@R0       ;SET CLOCK
1488 023336  042710  000002                  BIC     #MCLK,@R0       ;CLEAR CLOCKS
1489 023342  005302                        DEC     R2              ;COUND DOWN
1490 023344  001372                        BNE     12$             ;BRANCH IF 8 NOT DONE
1491 023346  052710  000012                BIS     #MSTCK!MCLK,@R0 ;SET SECTOR AND CLOCK
1492 023352  042710  000012                BIC     #MSTCK!MCLK,@R0 ;CLEAR SECTOR AND CLOCK
1493
1494                                ;NOW ALL REGISTERS WILL BE COMPARED
1495                                ;SO FILL EXPECTED VALUE INTO SAVED LOCATIONS
1496
1497 023356  052737  100000  002562        BIS     #SC,REINTO+6    ;INCLUDE SC IN SAVED RHCS1
1498 023364  053737  001416  002600        BIS     ATTENT,REINTO+24 ;FILL APPROPIATE ATTENTION
1499                                                                ;IN SAVED RHAS
1500 023372  052737  000001  002602        BIS     #DMD,REINTO+26  ;SET DMD IN RHMR SAVED
1501 023400  052737  100000  002604        BIS     #ATA,REINTO+30  ;SET ATA IN RHDS1 SAVED
1502 023406  013737  001200  002616        MOV     $TMP1,REINTO+42 ;MOVE EXPECTED VALUE
1503                                                                ;INTO RHLA SAVED
1504
1505                                ;AFTER SEARCH COMMAND
1506                                ;SAVE REGISTERS AGAIN SO THAT COMPARES CAN BE DONE
1507
1508 023414  004037  042152                JSR     R0,SAVER        ;SAVE
1509 023420  001232                        RHWC                    ;FROM
1510 023422  001510                        WRFROM                  ;TO
1511 023424  000023                        19.                     ;NUMBER
1512
1513                                ;AS UPPER BYTE OF RHAS CAN BE CHANGING IN A DUAL PORT
1514                                ;OPERATION THE UPPER BYTE OF RHAS WILL BE SAVED AS IS
1515                                ;SO THAT THE COMPARES ARE ONLY VALID FOR THE LOWER BYTE
1516 023426  113737  002601  001535        MOVB    REINTO+25,WRFROM+25;SAVE UPPER RHAS
1517
1518
1519                                ;COMPARE REGISTERS BEFORE SEARCH WITH AFTER
1520
1521 023434  004037  042354                JSR     R0,COMPAR       ;COMPAR
1522 023440  002554                        REINTO                  ;GO BUFFER
1523
1524 023442  001510                        WRFROM                  ;TEST BUFFER
1525 023444  000022                        18.                     ;NUMBER
1526 023446  023454                        13$                     ;RETURN FOR ERROR
1527 023450  023454                        13$                     ;SAME
1528 023452  023474                        14$                     ;RETURN FOR GOOD COMPARISON
1529 023454  013705  046426          13$:  MOV     ERWORD,R5       ;GETTING READY TO INDEX
1530 023460  060505                        ADD     R5,R5           ;DOUBLE ERROR WORD
1531 023462  016537  001230  041130        MOV     RHWC-2(R5),REGADR ;FAILING REG. ADDRESS
```

```
1532 023470  104001                    EMT     1
1533 023472  000207                    RTS     PC                  ;CHANGED AT END OF
1534 023474                         14$:                            ;SEARCH
1535                                ;;*******************************************************************
                                    ;*TEST 55        MAKE CURRENT CYLINDER = 0
                                    ;;*******************************************************************
     023474  000004                 TST55:  SCOPE
     023476  012706  001100                 MOV     #STACK,SP       ;RESET STACK
     023502  012737  000055  001432         MOV     #55,TSTNM       ;MOVE #55 TO TEST NUMBER
     023510  004737  041460                 JSR     PC,CLDISK       ;INIT DRIVE
     023514  012777  000001  155536         MOV     #DMD,@RHMR      ;SET DIAGNOSTIC MODE
     023522  004037  044032                 JSR     R0,MAKECYL      ;DO SEEK COMMAND FOLLOWED BY AN INIT TO
     023526  000000                         .WORD   0               ;CHANGE RHCC TO CYLINDER 0
```

```
                                .SBTTL  READ/WRITE ADDRESSING VIA RHMR

                                ;;*****************************************************************
                                ;*TEST 56       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)
                                ;;*****************************************************************
023530  000004                TST56:  SCOPE
023532  012706  001100                MOV     #STACK,SP       ;RESET STACK
023536  012737  000056  001432        MOV     #56,TSTNM       ;MOVE #56 TO TEST NUMBER

023544  012700  050126                MOV     #SECGAP,R0      ;POINTER
023550  012701  000460                MOV     #304.,R1        ;COUNTER
023554  012720  177777        1$:     MOV     #-1,(R0)+       ;CLEAR "DISK" AREA TO ALL ONES
023560  005301                        DEC     R1              ;
023562  001374                        BNE     1$              ;
023564  004737  041460                JSR     PC,CLDISK       ;THIS IS USED TO SET UP GENERAL
                                                              ;REGISTER CORRESPONDENCE

                                ;THESE ARE TO SET UP FOR DISKLESS USE ONLY

023570  012737  010000  051444        MOV     #FMT22,WCYL     ;FORMAT 22=16 BIT WORDS AND
                                                              ;CYLINDER 0
023576  005037  051446                CLR     WSECTR  ;TRACK=0, SECTOR=0
023602  005037  051450                CLR     WKEY1           ;KEY1=0
023606  005037  051452                CLR     WKEY2           ;KEY2=0
023612  012737  000400  051504        MOV     #256.,FNWORD    ;256 DATAWORDS
023620  004537  042666                JSR     R5,CRC  ;GO TO CALCULATE CRC
023624  051444                        WCYL
023626  051454                        GCRC

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

023630  012777  177374  155374        MOV     #-260.,@RHWC    ;256 DATA WORDS 4 HEADER WORDS
023636  012700  001510                MOV     #WRFROM,R0      ;BUS ADDRESS TO BE
023642  010077  155366                MOV     R0,@RHBA        ;BUFFER "WRFROM"
023646  012705  000403                MOV     #259.,R5        ;COUNTER
023652  012720  010000                MOV     #FMT22,(R0)+    ;FORMAT =16 BIT WORD

023656  005020                2$:     CLR     (R0)+           ;SECTOR=0, TRACK=0,KEYS=0, ALL DATA=0
023660  005305                        DEC     R5              ;& CYLINDER=0....SO CLEAR ALL "WRFROM"
023662  001375                        BNE     2$              ;CONTINUE IF ALL 259 NOT COMPLETE
023664  005077  155354                CLR     @RHDST          ;TRACK=0, SECTOR=0

023670  004737  041514                JSR     PC,CHECKT       ;CHECK THAT DVA,RDY,DPR,DRY = 1
023674  104401  062726                TYPE    ,CPHALT         ;AND THAT NO OTHERS = 1.  CANNOT CON-
                                                              ;TINUE TESTING IF BOTH AREN'T TRUE
023700  000000                        HALT                    ;STOP THE TEST

023702  013711  001464                MOV     WRIFOR,@R1      ;GET READY FOR WRITE HEADER
                                                              ;AND DATA WITH 62 IN RHCS1
023706  005037  001406                CLR     ERFLG$  ;CLEAR ERROR FLAG
023712  012777  010000  155330        MOV     #FMT22,@RHOF    ;FORMAT BIT=1 16 BIT WORDS
```

```
                                        CLR     @RHCA           ;CYLINDER 0
                                        JSR     PC,COMWHD       ;WRITE HEADER AND DATA FROM ''WRFROM''
                                                                ;INTO THE RHMR REGISTER AND BACK INTO
                                                                ;CORE ''DISK'' AREA

                                ;IF THE PROGRAM COMES BACK HERE WITHOUT ERROR
                                ;PRINT OUTS FROM THE ''COMWHD'' ROUTINE THAT MEANS
                                ;ALL HEADER ON DISK IS GOOD IE. ONLY DATA IS
                                ;TO BE CHECKED TO SEE IF IT IS ZERO

                                ;AND WRITE DATA GAP AND TOLERANCE GAP TO SEE IF THEY
                                ;ARE ALL ZEROS (ECC1 AND ECC2 MAY NOT BE 0) AS WELL AS
                                ;CHECKING RHWC FOR ZERO

023730  017737  155276  001126        MOV     @RHWC,$BDDAT    ;MOVE WORD COUNTER INTO BAD DATA
023736  001401                        BEQ     5$              ;SHOULD HAVE COUNTED UP TO ZERO
023740  104040                        EMT     40
                                                                ;HEADER AND DATA

023742  005737  001406          5$:   TST     ERFLG$  ;HAVE ANY ERRORS OCCURED?
023746  001034                        BNE     TST57           ;;BRANCH IF YES
023750  004737  041704                JSR     PC,CHECKE       ;CHECK THAT BITS = 1
023754  104401  062726                TYPE    ,CPHALT         ;CANNOT CONTINUE TESTING IF THEY DON'T
023760  000000                        HALT                    ;STOP THE TEST
023762  005037  051224                CLR     WECC1           ;CLEAR ECC
023766  005037  051226                CLR     WECC2

                                ;REINTO BUFFER IS FILLED WITH EXPECTED DATA OF ALL 0'S

023772  004037  041376                JSR     R0,CLAREA       ;CLEAR ''REINTO''
023776  002554                        REINTO                  ;FROM
024000  003614                        REINTO+<272.*2>         ;TO
024002  000000                .WORD   0                       ;FILL WITH ZEROS
024004  005037  001406                CLR     ERFLG$  ;CLEAR ERROR FLAG


                                ;COMPARE ''REINTO'' BUFFER WITH ''DISK'' BUFFER
024010  004037  042354                JSR     R0,COMPAR       ;CHECK
024014  002554                        REINTO                  ;GOOD BUFFER
024016  050224                        DISK                    ;TEST BUFFER
024020  000421                        273.                    ;NUMBER OF WORDS CHECKED
024022  024030                        3$                      ;RETURN POINT FOR ERROR HEADER
024024  024034                        4$                      ;RETURN POINT FOR ERROR DATA
024026  024040                        TST57                   ;RETURN FOR GOOD COMPARISON
024030                          3$:
024030  104007                        EMT     7
024032  000207                        RTS     PC              ;RETURN TO ''COMPAR''
024034                          4$:
024034  104010                        EMT     10
                                                                ;DATA WORDS
                                                                ;257 AND 258 ARE ECC
                                                                ;ZEROED OUT
                                                                ;259 TO 273 TOLERANCE GAP
024036  000207                        RTS     PC


                                ;;**********************************************************************
                                ;*TEST 57        WRITE HEADER AND DATA 2
```

```
T57     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)
                                        ;;*****************************************************************
    024040  000004                      TST57:  SCOPE
113 024042  012706  001100              MOV     #STACK,SP       ;RESET STACK
114 024046  012737  000057  001432      MOV     #57,TSTNM       ;MOVE #57 TO TEST NUMBER
115
116 024054  012700  050126              MOV     #SECGAP,R0      ;POINTER
117 024060  012701  000460              MOV     #304.,R1        ;COUNTER
118 024064  005020                      1$:     CLR     (R0)+           ;CLEAR SIMULATED "DISK" AREA IN CORE
119 024066  005301                      DEC     R1              ;
120 024070  001375                      BNE     1$              ;
121 024072  004737  041460              JSR     PC,CLDISK       ;THIS IS USED TO SET GENERAL REGISTERS
122
123                                     ;THESE ARE TO BE SETUP FOR DISKLESS USE ONLY
124
125 024076  012737  010000  051444      MOV     #FMT22,WCYL     ;FORMAT 22 = 16 BIT WORDS AND
126                                                             ;CYLINDER 0
127 024104  012737  000001  051446      MOV     #1,WSECTR       ;TRACK=0, SECTOR=1
128 024112  005037  051450              CLR     WKEY1           ;KEY1=0
129 024116  005037  051452              CLR     WKEY2           ;KEY2=0
130 024122  012737  000400  051504      MOV     #256.,FNWORD    ;256 DATA WORDS
131 024130  004537  042666              JSR     R5,CRC  ;GO TO CALCULATE CRC
132 024134  051444                      WCYL
133 024136  051454                      GCRC
134
135                                     ;THESE ARE REGULAR SETUPS FOR THE RH11 AND "WRFROM" BUFFER
136
137 024140  012777  177374  155064      MOV     #-260.,@RHWC    ;256 DATA WORDS 4 HEADER WORDS
138 024146  012700  001510              MOV     #WRFROM,R0      ;THESE TWO INSTRUCTIONS GETS
139 024152  010077  155056              MOV     R0,@RHBA        ;ADDR. OF WRFROM INTO R0 AND
140                                                             ;BUS ADDRESS REGISTER
141 024156  012720  010000              MOV     #FMT22,(R0)+    ;FORMAT=16 BIT WORDS
142                                                             ;CYLINDER=0
143 024162  012720  000001              2$:     MOV     #1,(R0)+        ;TRACK=0, SECTOR=1, KEYS=0
144 024166  005020                      CLR     (R0)+           ;KEY1=0
145 024170  005020                      CLR     (R0)+           ;KEY2=0
146 024172  012705  000400              MOV     #256.,R5        ;COUNTER
147 024176  012720  177777              3$:     MOV     #-1,(R0)+       ;MOVE ALL ONES FOR DATA
148 024202  005305                      DEC     R5
149 024204  001374                      BNE     3$              ;BRANCH IF DATA NOT COMPLETE
150 024206  012777  000001  155030      MOV     #1,@RHDST       ;TRACK=0 SECTOR=1
151
152 024214  004737  041514              JSR     PC,CHECKT       ;CHECK THAT DVA,RDY,DPR,DRY = 1
    024220  104401  062726              TYPE    ,CPHALT         ;AND THAT NO OTHERS = 1.  CANNOT CON-
                                                                ;TINUE TESTING IF BOTH AREN'T TRUE
    024224  000000                      HALT                    ;STOP THE TEST
153
154 024226  013711  001464              MOV     WRIFOR,@R1      ;GET READY FOR WRITE HEADER AND
155                                                             ;DATA WITH 62 IN RHCS1
156 024232  005037  001406              CLR     ERFLG$  ;CLEAR ERROR FLAG
157 024236  012777  010000  155004      MOV     #FMT22,@RHOF    ;FORMAT BIT=1 (16 BIT WORDS)
158 024244  005077  155002              CLR     @RHCA           ;CYLINDER =0
159 024250  004737  051270              JSR     PC,COMWHD       ;WRITE HEADER AND DATA INTO "DISK" AREA
```

```
                                        ;IF THE PROGRAM COMES BACK HERE WITHOUT ERROR PRINTOUTS
                                        ;FROM THE "COMWHD" ROUTINE THAT MEANS ALL THE HEADER ON "DISK"
                                        ;IS GOOD IE. ONLY THE DATA IS TO BE CHECKED TO SEE IF IT IS
                                        ;ALL ONES AND WRITE DATA GAP AND TOLERANCE GAP TO SEE IF
                                        ;THEY ARE ALL ZEROS, - ECC1 AND ECC2 ARE NOT CHECKED.

                                        ;RHWC IS CHECKED TO BE = 0

024254  017737  154752  001126            MOV     @RHWC,$BDDAT    ;LOAD WORD COUNTER JUST IN CASE
024262  001401                            BEQ     6$              ;SHOULD BE = 0
024264  104040                            EMT     40
                                                                  ;HEADER AND DATA IS COMPLETED

024266  005737  001406            6$:     TST     ERFLG$  ;HAVE ANY ERRORS OCCURRED?
024272  001041                            BNE     TST60           ;;BRANCH IF YES
024274  004737  041704                    JSR     PC,CHECKE       ;CHECK THAT BITS = 1
024300  104401  062726                    TYPE    ,CPHALT         ;CANNOT CONTINUE TESTING IF THEY DON'T
024304  000000                            HALT                    ;STOP THE TEST
024306  005037  051224                    CLR     WECC1           ;CLEAR ECC
024312  005037  051226                    CLR     WECC2           ;CLEAR ECC

                                        ;FILL "REINTO" BUFFER WITH EXPECTED DATA OF ALL 1'S

024316  004037  041376                    JSR     R0,CLAREA       ;FILL REINTO BUFFER
024322  002554                            REINTO                  ;FROM
024324  003552                            REINTO+<255.*2>         ;TO
024326  177777                    .WORD   -1                      ;DATA
024330  004037  041376                    JSR     R0,CLAREA       ;FILL REST
024334  003554                            REINTO+<256.*2>         ;FROM
024336  003614                            REINTO+<272.*2>         ;TO
024340  000000                            0                       ;DATA

024342  005037  001406                    CLR     ERFLG$  ;CLEAR ERROR FLAG

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

024346  004037  042354                    JSR     R0,COMPAR       ;CHECK
024352  002554                            REINTO                  ;GOOD BUFFER
024354  050224                            DISK                    ;TEST BUFFER
024356  000421                            273.                    ;NUMBER OF WORDS CHECKED
024360  024366                            4$                      ;RETURN POINT FOR ERROR HEADER
024362  024372                            5$                      ;RETURN POINT FOR ERROR DATA
024364  024376                            TST60                   ;RETURN FOR GOOD COMPARISON
024366                          4$:
024366  104007                            EMT     7
024370  000207                            RTS     PC              ;RETURN TO COMPARE
024372                          5$:
024372  104010                            EMT     10
                                                                  ;DATA WORDS
                                                                  ;WORD NOS 257 AND 258
                                                                  ;ARE ECC WHICH HAVE BEEN
                                                                  ;ZEROED
                                                                  ;WORD NOS 259
                                                                  ;IS DATA GAP
                                                                  ;WORD NOS 260 TO 273
                                                                  ;ARE TOLERANCE GAP
```

```
213 024374  000207                        RTS     PC              ;RETURN TO COMPARE
214
222                                 ;;*****************************************************************
                                    ;*TEST 60       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)
                                    ;;*****************************************************************
    024376  000004                  TST60:  SCOPE
223 024400  012706  001100                  MOV     #STACK,SP       ;RESET STACK
224 024404  012737  000060  001432          MOV     #60,TSTNM       ;MOVE #60 TO TEST NUMBER
225
226 024412  012700  050126                  MOV     #SECGAP,R0      ;POINTER
227 024416  012701  000460                  MOV     #304.,R1        ;COUNTER
228 024422  012720  000377          1$:     MOV     #377,(R0)+      ;LOAD SIMULATED "DISK" AREA WITH 377
229 024426  005301                          DEC     R1              ;
230 024430  001374                          BNE     1$              ;
231 024432  004737  041460                  JSR     PC,CLDISK       ;THIS IS USED TO SET UP GENERAL
232                                                                 ;REGISTER CORRESPONDENCE
233
234                                 ;THESE ARE TO BE SETUP FOR DISKLESS USE ONLY
235
236 024436  012737  010000  051444          MOV     #FMT22,WCYL     ;FORMAT 22=16 BIT WORDS AND
237                                                                 ;CYLINDER 0
238 024444  012737  000401  051446          MOV     #401,WSECTR     ;TRACK=1, SECTOR=1
239 024452  005037  051450                  CLR     WKEY1           ;KEY1=0
240 024456  005037  051452                  CLR     WKEY2           ;KEY2=0
241 024462  012737  000400  051504          MOV     #256.,FNWORD    ;256 DATA WORDS
242 024470  004537  042666                  JSR     R5,CRC  ;GO TO CALCULATE CRC
243 024474  051444                          WCYL
244 024476  051454                          GCRC
245
246                                 ;THESE ARE REGULAR SETUPS FOR RH11 AND "WRFROM" BUFFER
247
248 024500  012777  177374  154524          MOV     #-260.,@RHWC    ;256 DATA WORDS 4 HEADER WORDS
249 024506  012700  001510                  MOV     #WRFROM,R0      ;THESE TWO INSTRUCTIONS GET
250                                                                 ;ADDR. OF WRFROM INTO R0
251 024512  010077  154516                  MOV     R0,@RHBA        ;AND BUS ADDRESS REGISTER
252
253 024516  012720  010000                  MOV     #FMT22,(R0)+    ;FORMAT=16 BIT WORDS
254                                                                 ;CYLINDER=0
255 024522  012720  000401          2$:     MOV     #401,(R0)+      ;TRACK=1, SECTOR=1, KEYS=0
256 024526  005020                          CLR     (R0)+           ;KEY1=0
257 024530  005020                          CLR     (R0)+           ;KEY2=0
258 024532  012705  000400                  MOV     #256.,R5        ;COUNTER
259 024536  012720  052525          3$:     MOV     #052525,(R0)+   ;MOVE ALTERNATE ONES FOR DATA
260 024542  005305                          DEC     R5              ;COUNT
261 024544  001374                          BNE     3$              ;BRANCH IF DATA NOT COMPLETE
262 024546  012777  000401  154470          MOV     #401,@RHDST     ;TRACK=1 SECTOR=1
263
264 024554  004737  041514                  JSR     PC,CHECKT       ;CHECK THAT DVA,RDY,DPR,DRY = 1
    024560  104401  062726                  TYPE    ,CPHALT         ;AND THAT NO OTHERS = 1.  CANNOT CON-
                                                                    ;TINUE TESTING IF BOTH AREN'T TRUE
    024564  000000                          HALT                    ;STOP THE TEST
265
```

```
024566  013711  001464                  MOV     WRIFOR,@R1      ;GET READY FOR WRITE HEADER
                                                                ;AND DATA WITH 62 IN RHCS1
024572  005037  001406                  CLR     ERFLG$  ;CLEAR ERROR FLAG
024576  012777  010000  154444          MOV     #FMT22,@RHOF    ;FORMAT BIT=1(16 BIT WORDS
024604  005077  154442                  CLR     @RHCA           ;CYLINDER=0
024610  004737  051270                  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

024614  017737  154412  001126          MOV     @RHWC,$BDDAT    ;LOAD AND TEST FOR ZERO
024622  001401                          BEQ     6$              ;RHWC SHOULD = 0
024624  104040                          EMT     40
                                                                ;HEADER AND DATA WAS COMPLETED

                                ;ONLY ECC1 AND ECC2 ARE NOT CHECKED

024626  005737  001406          6$:     TST     ERFLG$  ;HAVE ANY ERRORS OCCURED?
024632  001041                          BNE     TST61           ;;BRANCH IF YES
024634  004737  041704                  JSR     PC,CHECKE       ;CHECK THAT BITS = 1
024640  104401  062726                  TYPE    ,CPHALT         ;CANNOT CONTINUE TESTING IF THEY DON'T
024644  000000                          HALT                    ;STOP THE TEST
024646  005037  051224                  CLR     WECC1           ;CLEAR ECC
024652  005037  051226                  CLR     WECC2           ;CLEAR ECC

                                ;FILL "REINTO" BUFFER WITH EXPECTED DATA

024656  004037  041376                  JSR     R0,CLAREA       ;FILL REINTO BUFFER
024662  002554                          REINTO                  ;FROM
024664  003552                          REINTO+<255.*2>         ;TO
024666  052525                          .WORD   52525           ;DATA
024670  004037  041376                  JSR     R0,CLAREA       ;FILL REST
024674  003554                                  REINTO+<256.*2> ;FROM
024676  003614                          REINTO+<272.*2>         ;TO
024700  000000                          .WORD   0               ;DATA
024702  005037  001406                  CLR     ERFLG$  ;CLEAR ERROR FLAG

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

024706  004037  042354                  JSR     R0,COMPAR       ;CHECK
024712  002554                          REINTO                  ;GOOD BUFFER
024714  050224                          DISK                    ;TEST BUFFER
024716  000421                          273.                    ;NUMBER OF WORDS CHECKED
024720  024726                          4$                      ;RETURN POINT FOR ERROR HEADER
024722  024732                          5$                      ;RETURN POINT FOR ERROR DATA
024724  024736                          TST61                   ;RETURN FOR GOOD COMPARISON
024726                          4$:
024726  104007                          EMT     7
024730  000207                          RTS     PC              ;RETURN TO COMPARE
024732                          5$:
024732  104010                          EMT     10
                                                                ;DATA WORDS
```

```
                                                        ;WORD NOS 257 AND 258
                                                        ;ARE ECC WHICH HAVE BEEN
                                                        ;ZEROED
                                                        ;WORD NOS 259
                                                        ;IS DATA GAP
                                                        ;WORD NOS 260 TO 273
                                                        ;ARE TOLERANCE GAP
024734  000207                  RTS     PC            ;RETURN TO COMPARE

                        ;;**************************************************************
                        ;*TEST 61        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
                        ;;**************************************************************
024736  000004         TST61:  SCOPE
024740  012706  001100          MOV     #STACK,SP     ;RESET STACK
024744  012737  000061  001432  MOV     #61,TSTNM     ;MOVE #61 TO TEST NUMBER

024752  012700  050126          MOV     #SECGAP,R0    ;POINTER
024756  012701  000460          MOV     #304.,R1      ;COUNTER
024762  012720  177777  1$:     MOV     #-1,(R0)+     ;CLEAR DISK AREA TO ALL ONES.
024766  005301                  DEC     R1            ;
024770  001374                  BNE     1$            ;
024772  004737  041460          JSR     PC,CLDISK     ;THIS IS USED TO SET GENERAL
                                                      ;REGISTERS

                        ;THESE ARE TO SET UP FOR DISKLESS USE ONLY

024776  012737  010000  051444  MOV     #FMT22,WCYL   ;FORMAT 22=16 BITWORDS AND
                                                      ;CYLINDER 0
025004  005037  051446          CLR     WSECTR  ;TRACK=0, SECTOR=0
025010  005037  051450          CLR     WKEY1         ;KEY1=0
025014  005037  051452          CLR     WKEY2         ;KEY2=0
025020  012737  000400  051504  MOV     #256.,FNWORD  ;256 DATAWORDS
025026  004537  042666          JSR     R5,CRC  ;GO TO CALCULATE CRC
025032  051444                  WCYL
025034  051454                  GCRC

                        ;THESE ARE REGULAR SETUPS

025036  012777  177374  154166  MOV     #-260.,@RHWC  ;256 DATA WORDS 4 HEADER WORDS
025044  012700  001510          MOV     #WRFROM,R0    ;FROM BUFFER 'WRFROM'
025050  010077  154160          MOV     R0,@RHBA      ;IN BUS ADDRESS
025054  012705  000403          MOV     #259.,R5      ;COUNTER
025060  012720  010000          MOV     #FMT22,(R0)+  ;FORMAT =16 BIT WORD
                                                      ;CYLINDER=0
025064  005020                  CLR     (R0)+         ;SECTOR=0, TRACK=0,KEYS=0, ALL DATA=0
025066  005305                  DEC     R5            ;COUNT
025070  001375                  BNE     2$            ;BRANCH IF ALL 259 NOT COMPLETE
025072  005077  154146          CLR     @RHDST        ;TRACK=0, SECTOR=0

025076  004737  041514          JSR     PC,CHECKT     ;CHECK THAT DVA,RDY,DPR,DRY = 1
025102  104401  062726          TYPE    ,CPHALT       ;AND THAT NO OTHERS = 1.  CANNOT CON-
                                                      ;TINUE TESTING IF BOTH AREN'T TRUE
025106  000000                  HALT                  ;STOP THE TEST
```

```
372
373 025110  013711  001464                  MOV     WRIFOR,@R1      ;GET READY FOR WRITE HEADER
374                                                                   ;AND DATA WITH 62 IN RHCS1
375 025114  005037  001406                  CLR     ERFLG$  ;CLEAR ERROR FLAG
376 025120  012777  010000  154122          MOV     #FMT22,@RHOF    ;FORMAT BIT=1 16 BIT WORDS
377 025126  005077  154120                  CLR     @RHCA           ;CYLINDER 0
378 025132  012777  000001  154120          MOV     #DMD,@RHMR      ;SET DIAGNOSTIC MODE
379 025140  052777  000001  154072          BIS     #GO,@RHCS1      ;GO
380 025146  000240                          NOP
381 025150  052777  000001  154062          BIS     #GO,@RHCS1      ;THIS GO SHOULD SET PGE
382
383 025156  004737  041064                  JSR     PC,PUTREG       ;SAVE REGISTERS
384 025162  032737  002000  001314          BIT     #PGE,CS1        ;IS PGE SET
385 025170  001404                          BEQ     3$              ;BRANCH IF GOOD
386 025172  013737  001236  001122          MOV     RHCS2,$BDADR
387 025200  104037                          EMT     37
388                                                                   ;WAS ATTEMPTED WITH ONE IN PROGRESS
389 025202                          3$:
```

```
                                .SBTTL  MAINTAINANCE REGISTER TESTS THRU RHMR

                                ;;*****************************************************************
                                ;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:

                                ;THE HEADER CONSISTING OF CYLINDER ADDRESS, SECTOR,
                                ;TRACK AND THE KEYS ARE READ FROM LOCATION
                                ;CYL:, SECTOR:, KEY1:, AND KEY2 AND NOT FROM
                                ;HEADER: ON SIMULATED DISK

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

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

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

                                ;;*****************************************************************
                                ;*TEST 62        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)
                                ;;*****************************************************************
   025202  000004               TST62:  SCOPE
30 025204  012706  001100               MOV     #STACK,SP       ;RESET STACK
31 025210  012737  000062  001432       MOV     #62,TSTNM       ;MOVE #62 TO TEST NUMBER
32
33
                                ;SETUP FOR WHAT IS TO BE READ
                                ;HEADER CRC IS RESTORED FROM A SUBROUTINE

   025216  012746  000000               MOV     #0,-(SP)        ;DATA TO BE READ
   025222  012705  000400               MOV     #256.,R5        ;COUNTER
   025226  012700  050224               MOV     #DISK,R0        ;START OF SIMULATED DISK DATA
   025232  011620               1$:     MOV     (SP),(R0)+      ;MOVE IN DATA ON TO SIMULATED DISK
   025234  005305                       DEC     R5              ;COUNT
   025236  001375                       BNE     1$              ;BRANCH IF 256 NOT COMPLETE
   025240  005726                       TST     (SP)+           ;UNDO -(SP)
   025242  012705  000021               MOV     #17.,R5         ;2 ECC WORDS
                                                                ;1 DATA GAP
                                                                ;14 TOLERANCE GAP
   025246  005020               2$:     CLR     (R0)+           ;CLEAR ECC, DATA GAP, AND
   025250  005305                       DEC     R5              ;TOLERANCE GAP
   025252  001375                       BNE     2$              ;BRANCH IF NOT COMPLETE

                                ;THESE ARE TO SETUP FOR DISKLESS USE ONLY

   025254  012737  010000  046306       MOV     #0!FMT22,CYL    ;16 BITS PER WORD
                                                                ;CYLINDER 0, FORMAT 16 BITS
   025262  112737  000000  046311       MOVB    #0,SECOTR+1     ;TRACK 0
   025270  112737  000000  046310       MOVB    #0,SECOTR       ;SECTOR 0
```

```
025276  012737  000000  046312          MOV     #0,KEY1 ;KEY1=0
025304  012737  000000  046314          MOV     #0,KEY2 ;KEY2=0
025312  012737  000400  046366          MOV     #256.,DAWORD    ;NO. OF DATA WORDS
025320  005037  046316                  CLR     X               ;THIS IS A READ COMMAND
025324  004537  042666                  JSR     R5,CRC          ;GO TO CALCULATE CRC
025330  046306                          CYL
025332  050206                          WCRC

                                ;THESE ARE REGULAR SETUPS FOR RH11 AND "REINTO"

025334  004737  041460                  JSR     PC,CLDISK       ;SETUP GENERAL REGISTERS
025340  012777  177374  153664          MOV     #-256.-4.,@RHWC ;256. DATA 4 HEADER WORDS
025346  012777  002554  153660          MOV     #REINTO,@RHBA   ;STARTING ADDRESS OF READ BUFFER
025354  112746  000000                  MOVB    #0,-(SP)        ;IN LOWER BYTE GET SECTOR
025360  112766  000000  000001          MOVB    #0,1(SP)        ;GET TRACK IN HIGHER BYTE
025366  012677  153652                  MOV     (SP)+,@RHDST    ;TRACK/SECTOR IN RHDST
025372  012777  014000  153650          MOV     #FMT22!ECI,@RHOF ;16 BITS PER WORD
                                                                ;ECC CORRECTION INHIBIT
                                                                ;BECAUSE ECC IS NOT GOING
                                                                ;TO BE CHECKED
025400  005077  153646                  CLR     @RHCA           ;CYLINDER 0

025404  004737  041514                  JSR     PC,CHECKT       ;CHECK THAT DVA,RDY,DPR,DRY = 1
025410  104401  062726                  TYPE    ,CPHALT         ;AND THAT NO OTHERS = 1.  CANNOT CON-
                                                                ;TINUE TESTING IF BOTH AREN'T TRUE
025414  000000                          HALT                    ;STOP THE TEST

025416  013711  001470                  MOV     REFOR,@R1       ;READ HEADER AND DATA=72
025422  005037  001406                  CLR     ERFLG$          ;CLEAR ERROR FLAG
025426  004737  046146                  JSR     PC,COMHD        ;READ HEADER AND DATA


                                ;IF THE PROGRAM COMES BACK HERE WITHOUT ERROR PRINTOUTS
                                ;FROM THE "COMHD" ROUTINE THAT MEANS SECTOR GAP,
                                ;FIRST SYNC, HEADER, HEADER CRC, HEADER GAP AND
                                ;SYNC BYTE HAVE GONE BY AND SYNCS WERE CORRECTLY
                                ;DETECTED.

                                ;RHWC IS CHECKED TO BE = 0 AFTER THE READ OPERATION

025432  017737  153574  001126          MOV     @RHWC,$BDDAT    ;LOAD AND TEST RHWC
025440  001401                          BEQ     20$             ;SHOULD = 0
025442  104040                          EMT     40
                                                                ;HEADER AND DATA

                                ;HEADER AND DATA ARE TO BE CHECKED.
                                ;IN CHECKING READ DATA THE WRITE FROM BUFFER
                                ;"WRFROM" IS FILLED WITH EXPECTED DATA AND
                                ;COMPARISONS ARE MADE.

025444  005737  001406          20$:    TST     ERFLG$          ;ANY ERRORS ALREADY THERE
025450  001046                          BNE     TST63           ;BRANCH IF YES
025452  004737  041704                  JSR     PC,CHECKE       ;CHECK THAT BITS = 1
025456  104401  062726                  TYPE    ,CPHALT         ;CANNOT CONTINUE TESTING IF THEY DON'T
025462  000000                          HALT                    ;STOP THE TEST
025464  012700  001510                  MOV     #WRFROM,R0      ;GETTING READY TO FILL EXPECTED DATA
025470  012720  010000                  MOV     #0!FMT22,(R0)+ ;CYLINDER 0
```

```
025474  112746  000000                  MOVB    #0,-(SP)        ;IN LOWER BYTE GET SECTOR
025500  112766  000000  000001          MOVB    #0,1(SP)        ;GET TRACK IN HIGHER BYTE
025506  012620                          MOV     (SP)+,(R0)+     ;GET TRACK/SECTOR IN BUFFER
025510  012720  000000                  MOV     #0,(R0)+        ;KEY1 IN BUFFER
025514  012720  000000                  MOV     #0,(R0)+        ;KEY2 IN BUFFER
025520  012701  000400                  MOV     #256.,R1        ;DATA WORD COUNTER
025524  012702  000000                  MOV     #0,R2           ;DATA
025530  010220                  3$:     MOV     R2,(R0)+        ;DATA INTO BUFFER
025532  005301                          DEC     R1              ;COUNT
025534  001375                          BNE     3$              ;BRANCH IF 256 NOT DONE

                        ;NOW READ DATA BUFFER WILL BE CHECKED

025536  004037  042354                  JSR     R0,COMPAR       ;CHECK
025542  001510                          WRFROM                  ;GOOD BUFFER
025544  002554                          REINTO                  ;TEST BUFFER
025546  000404                          4+256.                  ;NUMBER OF WORDS CHECKED
025550  025556                          4$                      ;RETURN POINT FOR ERROR HEADER
025552  025562                          5$                      ;RETURN POINT FOR ERROR DATA
025554  025566                          TST63                   ;RETURN FOR GOOD COMPARISON
025556                          4$:
025556  104004                          EMT     4
025560  000207                          RTS     PC              ;RETURN TO ''COMPAR''
025562                          5$:
025562  104005                          EMT     5
                                                                ;HEADER WORDS
                                                                ;5 TO 260 ARE DATA WORDS
025564  000207                          RTS     PC              ;RETURN TO ''COMPAR''

34
41
                        ;;*******************************************************************
                        ;*TEST 63       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)
                        ;;*******************************************************************
025566  000004         TST63:  SCOPE
42 025570  012706  001100                  MOV     #STACK,SP       ;RESET STACK
43 025574  012737  000063  001432          MOV     #63,TSTNM       ;MOVE #63 TO TEST NUMBER
44
45

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

025602  012746  177777                  MOV     #-1,-(SP)       ;DATA TO BE READ
025606  012705  000400                  MOV     #256.,R5        ;COUNTER
025612  012700  050224                  MOV     #DISK,R0        ;START OF SIMULATED DISK DATA
025616  011620                  1$:     MOV     (SP),(R0)+      ;MOVE IN DATA ON TO SIMULATED DISK
025620  005305                          DEC     R5              ;COUNT
025622  001375                          BNE     1$              ;BRANCH IF 256 NOT COMPLETE
025624  005726                          TST     (SP)+           ;UNDO -(SP)
025626  012705  000021                  MOV     #17.,R5         ;2 ECC WORDS
                                                                ;1 DATA GAP
                                                                ;14 TOLERANCE GAP
025632  005020                  2$:     CLR     (R0)+           ;CLEAR ECC, DATA GAP, AND
025634  005305                          DEC     R5              ;TOLERANCE GAP
```

```
025636  001375                          BNE     2$              ;BRANCH IF NOT COMPLETE

                                ;THESE ARE TO SETUP FOR DISKLESS USE ONLY

025640  012737  010000  046306          MOV     #0!FMT22,CYL    ;16 BITS PER WORD
                                                                ;CYLINDER 0, FORMAT 16 BITS
025646  112737  000000  046311          MOVB    #0,SECOTR+1     ;TRACK 0
025654  112737  000001  046310          MOVB    #1,SECOTR       ;SECTOR 1
025662  012737  000000  046312          MOV     #0,KEY1 ;KEY1=0
025670  012737  000000  046314          MOV     #0,KEY2 ;KEY2=0
025676  012737  000400  046366          MOV     #256.,DAWORD    ;NO. OF DATA WORDS
025704  005037  046316                  CLR     X               ;THIS IS A READ COMMAND
025710  004537  042666                  JSR     R5,CRC          ;GO TO CALCULATE CRC
025714  046306                          CYL
025716  050206                          WCRC

                                ;THESE ARE REGULAR SETUPS FOR RH11 AND "REINTO"

025720  004737  041460                  JSR     PC,CLDISK       ;SETUP GENERAL REGISTERS
025724  012777  177374  153300          MOV     #-256.-4.,@RHWC ;256. DATA 4 HEADER WORDS
025732  012777  002554  153274          MOV     #REINTO,@RHBA   ;STARTING ADDRESS OF READ BUFFER
025740  112746  000001                  MOVB    #1,-(SP)        ;IN LOWER BYTE GET SECTOR
025744  112766  000000  000001          MOVB    #0,1(SP)        ;GET TRACK IN HIGHER BYTE
025752  012677  153266                  MOV     (SP)+,@RHDST    ;TRACK/SECTOR IN RHDST
025756  012777  014000  153264          MOV     #FMT22!ECI,@RHOF ;16 BITS PER WORD
                                                                ;ECC CORRECTION INHIBIT
                                                                ;BECAUSE ECC IS NOT GOING
                                                                ;TO BE CHECKED
025764  005077  153262                  CLR     @RHCA           ;CYLINDER 0

025770  004737  041514                  JSR     PC,CHECKT       ;CHECK THAT DVA,RDY,DPR,DRY = 1
025774  104401  062726                  TYPE    ,CPHALT         ;AND THAT NO OTHERS = 1.  CANNOT CON-
                                                                ;TINUE TESTING IF BOTH AREN'T TRUE
026000  000000                          HALT                    ;STOP THE TEST

026002  013711  001470                  MOV     REFOR,@R1       ;READ HEADER AND DATA=72
026006  005037  001406                  CLR     ERFLG$          ;CLEAR ERROR FLAG
026012  004737  046146                  JSR     PC,COMHD        ;READ HEADER AND DATA


                                ;IF THE PROGRAM COMES BACK HERE WITHOUT ERROR PRINTOUTS
                                ;FROM THE "COMHD" ROUTINE THAT MEANS SECTOR GAP,
                                ;FIRST SYNC, HEADER, HEADER CRC, HEADER GAP AND
                                ;SYNC BYTE HAVE GONE BY AND SYNCS WERE CORRECTLY
                                ;DETECTED.

                                ;RHWC IS CHECKED TO BE = 0 AFTER THE READ OPERATION

026016  017737  153210  001126          MOV     @RHWC,$BDDAT    ;LOAD AND TEST RHWC
026024  001401                          BEQ     20$             ;SHOULD = 0
026026  104040                          EMT     40
                                                                ;HEADER AND DATA

                                ;HEADER AND DATA ARE TO BE CHECKED.
                                ;IN CHECKING READ DATA THE WRITE FROM BUFFER
                                ;"WRFROM" IS FILLED WITH EXPECTED DATA AND
                                ;COMPARISONS ARE MADE.
```

```
026030  005737  001406          20$:    TST     ERFLG$          ;ANY ERRORS ALREADY THERE
026034  001046                          BNE     TST64           ;BRANCH IF YES
026036  004737  041704                  JSR     PC,CHECKE       ;CHECK THAT BITS = 1
026042  104401  062726                  TYPE    ,CPHALT         ;CANNOT CONTINUE TESTING IF THEY DON'T
026046  000000                          HALT                    ;STOP THE TEST
026050  012700  001510                  MOV     #WRFROM,R0      ;GETTING READY TO FILL EXPECTED DATA
026054  012720  010000                  MOV     #0!FMT22,(R0)+ ;CYLINDER 0
026060  112746  000001                  MOVB    #1,-(SP)        ;IN LOWER BYTE GET SECTOR
026064  112766  000000  000001          MOVB    #0,1(SP)        ;GET TRACK IN HIGHER BYTE
026072  012620                          MOV     (SP)+,(R0)+     ;GET TRACK/SECTOR IN BUFFER
026074  012720  000000                  MOV     #0,(R0)+        ;KEY1 IN BUFFER
026100  012720  000000                  MOV     #0,(R0)+        ;KEY2 IN BUFFER
026104  012701  000400                  MOV     #256.,R1        ;DATA WORD COUNTER
026110  012702  177777                  MOV     #-1,R2          ;DATA
026114  010220                  3$:     MOV     R2,(R0)+        ;DATA INTO BUFFER
026116  005301                          DEC     R1              ;COUNT
026120  001375                          BNE     3$              ;BRANCH IF 256 NOT DONE

                                ;NOW READ DATA BUFFER WILL BE CHECKED

026122  004037  042354                  JSR     R0,COMPAR       ;CHECK
026126  001510                          WRFROM                  ;GOOD BUFFER
026130  002554                          REINTO                  ;TEST BUFFER
026132  000404                          4+256.                  ;NUMBER OF WORDS CHECKED
026134  026142                          4$                      ;RETURN POINT FOR ERROR HEADER
026136  026146                          5$                      ;RETURN POINT FOR ERROR DATA
026140  026152                          TST64                   ;RETURN FOR GOOD COMPARISON
026142                          4$:
026142  104004                          EMT     4
026144  000207                          RTS     PC              ;RETURN TO ''COMPAR''
026146                          5$:
026146  104005                          EMT     5
                                                                ;HEADER WORDS
                                                                ;5 TO 260 ARE DATA WORDS
026150  000207                          RTS     PC              ;RETURN TO ''COMPAR''

46
53                              ;;*****************************************************************
                                ;*TEST 64        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)
                                ;;*****************************************************************
026152  000004                  TST64:  SCOPE
54 026154  012706  001100               MOV     #STACK,SP       ;RESET STACK
55 026160  012737  000064  001432       MOV     #64,TSTNM       ;MOVE #64 TO TEST NUMBER
56
57

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

026166  012746  052525                  MOV     #052525,-(SP)   ;DATA TO BE READ
026172  012705  000400                  MOV     #256.,R5        ;COUNTER
026176  012700  050224                  MOV     #DISK,R0        ;START OF SIMULATED DISK DATA
026202  011620                  1$:     MOV     (SP),(R0)+      ;MOVE IN DATA ON TO SIMULATED DISK
```

```
026204  005305                          DEC     R5              ;COUNT
026206  001375                          BNE     1$              ;BRANCH IF 256 NOT COMPLETE
026210  005726                          TST     (SP)+           ;UNDO -(SP)
026212  012705  000021                  MOV     #17.,R5         ;2 ECC WORDS
                                                                ;1 DATA GAP
                                                                ;14 TOLERANCE GAP
026216  005020                  2$:     CLR     (R0)+           ;CLEAR ECC, DATA GAP, AND
026220  005305                          DEC     R5              ;TOLERANCE GAP
026222  001375                          BNE     2$              ;BRANCH IF NOT COMPLETE

                                ;THESE ARE TO SETUP FOR DISKLESS USE ONLY

026224  012737  010000  046306          MOV     #0!FMT22,CYL    ;16 BITS PER WORD
                                                                ;CYLINDER 0, FORMAT 16 BITS
026232  112737  000001  046311          MOVB    #1,SECOTR+1     ;TRACK 1
026240  112737  000001  046310          MOVB    #1,SECOTR       ;SECTOR 1
026246  012737  000000  046312          MOV     #0,KEY1 ;KEY1=0
026254  012737  000000  046314          MOV     #0,KEY2 ;KEY2=0
026262  012737  000400  046366          MOV     #256.,DAWORD    ;NO. OF DATA WORDS
026270  005037  046316                  CLR     X               ;THIS IS A READ COMMAND
026274  004537  042666                  JSR     R5,CRC          ;GO TO CALCULATE CRC
026300  046306                          CYL
026302  050206                          WCRC

                                ;THESE ARE REGULAR SETUPS FOR RH11 AND "REINTO"

026304  004737  041460                  JSR     PC,CLDISK       ;SETUP GENERAL REGISTERS
026310  012777  177374  152714          MOV     #-256.-4.,@RHWC ;256. DATA 4 HEADER WORDS
026316  012777  002554  152710          MOV     #REINTO,@RHBA   ;STARTING ADDRESS OF READ BUFFER
026324  112746  000001                  MOVB    #1,-(SP)        ;IN LOWER BYTE GET SECTOR
026330  112766  000001  000001          MOVB    #1,1(SP)        ;GET TRACK IN HIGHER BYTE
026336  012677  152702                  MOV     (SP)+,@RHDST    ;TRACK/SECTOR IN RHDST
026342  012777  014000  152700          MOV     #FMT22!ECI,@RHOF ;16 BITS PER WORD
                                                                ;ECC CORRECTION INHIBIT
                                                                ;BECAUSE ECC IS NOT GOING
                                                                ;TO BE CHECKED
026350  005077  152676                  CLR     @RHCA           ;CYLINDER 0

026354  004737  041514                  JSR     PC,CHECKT       ;CHECK THAT DVA,RDY,DPR,DRY = 1
026360  104401  062726                  TYPE    ,CPHALT         ;AND THAT NO OTHERS = 1.  CANNOT CON-
                                                                ;TINUE TESTING IF BOTH AREN'T TRUE
026364  000000                          HALT                    ;STOP THE TEST

026366  013711  001470                  MOV     REFOR,@R1       ;READ HEADER AND DATA=72
026372  005037  001406                  CLR     ERFLG$          ;CLEAR ERROR FLAG
026376  004737  046146                  JSR     PC,COMHD        ;READ HEADER AND DATA


                        ;IF THE PROGRAM COMES BACK HERE WITHOUT ERROR PRINTOUTS
                        ;FROM THE "COMHD" ROUTINE THAT MEANS SECTOR GAP,
                        ;FIRST SYNC, HEADER, HEADER CRC, HEADER GAP AND
                        ;SYNC BYTE HAVE GONE BY AND SYNCS WERE CORRECTLY
                        ;DETECTED.

                        ;RHWC IS CHECKED TO BE = 0 AFTER THE READ OPERATION

026402  017737  152624  001126          MOV     @RHWC,$BDDAT    ;LOAD AND TEST RHWC
```

```
026410  001401                          BEQ     20$             ;SHOULD = 0
026412  104040                          EMT     40
                                                                ;HEADER AND DATA

                                ;HEADER AND DATA ARE TO BE CHECKED.
                                ;IN CHECKING READ DATA THE WRITE FROM BUFFER
                                ;"WRFROM" IS FILLED WITH EXPECTED DATA AND
                                ;COMPARISONS ARE MADE.

026414  005737  001406          20$:    TST     ERFLG$          ;ANY ERRORS ALREADY THERE
026420  001046                          BNE     TST65           ;BRANCH IF YES
026422  004737  041704                  JSR     PC,CHECKE       ;CHECK THAT BITS = 1
026426  104401  062726                  TYPE    ,CPHALT         ;CANNOT CONTINUE TESTING IF THEY DON'T
026432  000000                          HALT                    ;STOP THE TEST
026434  012700  001510                  MOV     #WRFROM,R0      ;GETTING READY TO FILL EXPECTED DATA
026440  012720  010000                  MOV     #0!FMT22.,(R0)+ ;CYLINDER 0
026444  112746  000001                  MOVB    #1,-(SP)        ;IN LOWER BYTE GET SECTOR
026450  112766  000001  000001          MOVB    #1,1(SP)        ;GET TRACK IN HIGHER BYTE
026456  012620                          MOV     (SP)+,(R0)+     ;GET TRACK/SECTOR IN BUFFER
026460  012720  000000                  MOV     #0,(R0)+        ;KEY1 IN BUFFER
026464  012720  000000                  MOV     #0,(R0)+        ;KEY2 IN BUFFER
026470  012701  000400                  MOV     #256.,R1        ;DATA WORD COUNTER
026474  012702  052525                  MOV     #052525,R2              ;DATA
026500  010220                  3$:     MOV     R2,(R0)+        ;DATA INTO BUFFER
026502  005301                          DEC     R1              ;COUNT
026504  001375                          BNE     3$              ;BRANCH IF 256 NOT DONE

                                ;NOW READ DATA BUFFER WILL BE CHECKED

026506  004037  042354                  JSR     R0,COMPAR       ;CHECK
026512  001510                          WRFROM                  ;GOOD BUFFER
026514  002554                          REINTO                  ;TEST BUFFER
026516  000404                          4+256.                  ;NUMBER OF WORDS CHECKED
026520  026526                          4$                      ;RETURN POINT FOR ERROR HEADER
026522  026532                          5$                      ;RETURN POINT FOR ERROR DATA
026524  026536                          TST65                   ;RETURN FOR GOOD COMPARISON
026526                          4$:
026526  104004                          EMT     4
026530  000207                          RTS     PC              ;RETURN TO "COMPAR"
026532                          5$:
026532  104005                          EMT     5
                                                                ;HEADER WORDS
                                                                ;5 TO 260 ARE DATA WORDS
026534  000207                          RTS     PC              ;RETURN TO "COMPAR"

58
65
                                ;;*****************************************************************
                                ;*TEST 65        WRITE DATA
                                ;*WRITE CYLINDER 0, FORMAT 16 BITS PER WORD
                                ;*TRACK 0, SECTOR 0, KEYS 0, NUMBER OF WORDS 256 OF 377
                                ;*ANY ERROR LOGIC INDICATION IS NOT CONCLUSIVE ON FIRST PASS
                                ;*BECAUSE ERROR LOGIC HAS NOT BEEN CHECKED
                                ;;*****************************************************************
026536  000004                  TST65:  SCOPE
66 026540  012737  000065  001432       MOV     #65,TSTNM       ;MOVE #65 TO TEST NUMBER
67
68 026546  012706  001100               MOV     #STACK,SP       ;RESET STACK
```

```
026552  012737  000065  001432          MOV     #65,TSTNM       ;MOVE #65 TO TEST NUMBER

026560  004037  041376                  JSR     R0,CLAREA       ;CLEAR SIMULATED DISK
026564  050224                          .WORD   DISK            ;FROM
026566  051250                          .WORD   TOLGAP+16       ;TO
026570  000000                          .WORD   0               ;DATA

                                ;THESE ARE SETUP FOR DISKLESS USE ONLY

026572  012737  010000  046306          MOV     #0!FMT22,CYL    ;CYLINDER 0
                                                                ;16 BITS PER WORD
026600  112737  000000  046311          MOVB    #0,SECOTR+1     ;TRACK 0
026606  112737  000000  046310          MOVB    #0,SECOTR       ;SECTOR 0
026614  005037  046312                  CLR     KEY1            ;KEY1 0
026620  005037  046314                  CLR     KEY2            ;KEY2 0
026624  012737  000400  046354          MOV     #256.,NOWORD    ;NO OF DATA WORDS
026632  012737  000001  046316          MOV     #1,X            ;WRITE DATA
026640  004537  042666                  JSR     R5,CRC          ;GO TO CALCULATE CRC
026644  046306                          CYL
026646  050206                          WCRC

                                ;THESE ARE REGULAR SETUPS

026650  004037  041376                  JSR     R0,CLAREA       ;FILL WRITE BUFFER WITH 377
026654  001510                          WRFROM                  ;FROM LOCATION
026656  002510                          WRFROM+<256.*2>         ;TO LOCATION
026660  000377                          377                     ;DATA
026662  004737  041460                  JSR     PC,CLDISK       ;SETUP GENERAL REGISTERS
026666  012777  177400  152336          MOV     #-256.,@RHWC    ;256. DATA WORDS
026674  012777  001510  152332          MOV     #WRFROM,@RHBA   ;STARTING ADDRESS OF WRITE BUFFER
026702  012746  000000                  MOV     #0,-(SP)        ;SECTOR 0
026706  112766  000000  000001          MOVB    #0,1(SP)        ;TRACK 0
026714  012677  152324                  MOV     (SP)+,@RHDST    ;SECTOR 0 TRACK 0
026720  012777  010000  152322          MOV     #FMT22,@RHOF    ;16 BITS PER WORD FORMAT
026726  012777  000000  152316          MOV     #0,@RHCA        ;CYLINDER 0
026734  004737  041514                  JSR     PC,CHECKT       ;CHECK THAT DVA,RDY,DPR,DRY = 1
026740  104401  062726                  TYPE    ,CPHALT         ;AND THAT NO OTHERS = 1.  CANNOT CON-
                                                                ;TINUE TESTING IF BOTH AREN'T TRUE
026744  000000                          HALT                    ;STOP THE TEST
026746  013711  001462                  MOV     WRIDAT,@R1      ;WRITE DATA=60
026752  005037  001406                  CLR     ERFLG$          ;CLEAR ERROR FLAG
026756  004737  046146                  JSR     PC,COMHD        ;WRITE DATA

                                ;IF THE PROGRAM COMES BACK HERE WITHOUT ERROR PRINTOUTS
                                ;FROM THE "COMHD" ROUTINE IT MEANS SECTOR GAP, SYNC BYTE
                                ;HEADER, HEADER CRC, HEADER GAP AND SYNC BYTE HAVE GONE BY
                                ;AND SYNCS WERE CORRECTLY DETECTED, DATA IS TO BE CHECKED.

026762  004737  041064                  JSR     PC,PUTREG       ;SAVE REGISTERS
026766  005737  001406                  TST     ERFLG$          ;HAVE ANY ERRORS OCCURED?
026772  001041                          BNE     TST66           ;BRANCH IF YES
026774  012700  000377                  MOV     #377,R0         ;GOOD DATA
027000  012701  050224                  MOV     #DISK,R1        ;DATA WRITTEN INTO "DISK"
027004  012702  000400                  MOV     #256.,R2        ;COUNTER

027010  012737  000401  046426  1$:     MOV     #256.+1,ERWORD  ;FOR ERROR WORD
027016  020021                          CMP     R0,(R1)+        ;COMPARE GOOD DATA WITH DATA ON DISK
```

```
027020  001424                          BEQ     3$              ;BRANCH IF GOOD
027022  010037  001124                  MOV     R0,$GDDAT       ;GOOD DATA
027026  014137  001126                  MOV     -(R1),$BDDAT    ;BAD DATA
027032  160237  046426                  SUB     R2,ERWORD       ;ERROR WORD NO
027036  005737  001406                  TST     ERFLG$          ;ANY ERRORS ALREADY THERE?
027042  001002                          BNE     2$              ;BRANCH IF YES
027044  104004                          EMT     4
027046  000401                          BR      1064$           ;BRANCH TO AVOID PRINTING NEXT ERROR

027050                          2$:
027050  104005                          EMT     5
027052  005721                  1064$:  TST     (R1)+           ;UNDO -(R1) FOR BAD DATA
027054  017746  152060                  MOV     @SWR,-(SP)      ;GET SWITCH SETTING
027060  042716  177177                  BIC     #177177,(SP)    ;KEEP ONLY SWITCH 7 AND 8
027064  022726  000200                  CMP     #SW07,(SP)+     ;IS 7 SET AND 8 RESET
027070  001402                          BEQ     TST66           ;BRANCH OUT IF YES
027072  005302                  3$:     DEC     R2              ;IF NOT COUNT 256 WORDS
027074  001345                          BNE     1$              ;BRANCH IF 256. NOT DONE

 69
 76                                     ;;*****************************************************************
                                        ;*TEST 66       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)
                                        ;;*****************************************************************
027076  000004                  TST66:  SCOPE
 77 027100  012706  001100              MOV     #STACK,SP       ;RESET STACK
 78
 79 027104  012737  000066  001432      MOV     #66,TSTNM       ;MOVE #66 TO TEST NUMBER
 80 027112  004037  041376              JSR     R0,CLAREA       ;CLEAR SIMULATED DISK
 81 027116  050224                      .WORD   DISK            ;FROM
 82 027120  051222                      .WORD   DISK+776        ;TO
 83 027122  177400                      .WORD   177400          ;DATA
 84
 85 027124  004037  041376              JSR     R0,CLAREA       ;CLEAR READ INTO BUFFER
 86 027130  002554                      .WORD   REINTO          ;FROM
 87 027132  003552                      .WORD   REINTO+776      ;TO
 88 027134  000000                      .WORD   0               ;DATA
 89
 90                                 ;THESE ARE TO SETUP FOR DISKLESS USE ONLY
 91
 92 027136  012737  010000  046306      MOV     #FMT22,CYL      ;CYLINDER 0 16 BITS PER WORD FORMAT
 93 027144  105037  046311              CLRB    SECOTR+1        ;TRACK 0
 94 027150  112737  000001  046310      MOVB    #1,SECOTR       ;SECTOR 1
 95 027156  005037  046312              CLR     KEY1            ;KEY1=0
 96 027162  005037  046314              CLR     KEY2            ;KEY2=0
 97 027166  012737  000012  046366      MOV     #10.,DAWORD     ;NO. OF DATA WORDS
 98 027174  005037  046316              CLR     X               ;THIS IS A READ COMMAND
 99 027200  004537  042666              JSR     R5,CRC  ;GO TO CALCULATE CRC
100 027204  046306                      CYL
101 027206  050206                      WCRC
102
103                                ;THESE ARE REGULAR SETUPS
104
105 027210  004737  041460              JSR     PC,CLDISK       ;SETUP GENERAL REGISTERS
```

```
    106 027214  013711  001466                  MOV     READAT,@R1      ;READ DATA INTO RHCS1=70
    107 027220  012777  177766  152004          MOV     #-10.,@RHWC     ;10 DATA WORDS
    108 027226  012777  002554  152000          MOV     #REINTO,@RHBA   ;STARTING ADDRESS OF READ BUFFER
    109 027234  112746  000001                  MOVB    #1,-(SP)        ;IN LOWER BYTE GET SECTOR 1
    110 027240  112766  000000  000001          MOVB    #0,1(SP)        ;GET TRACK0 IN UPPER BYTE
    111 027246  012677  151772                  MOV     (SP)+,@RHDST    ;TRACK/SECTOR IN RHDST
    112 027252  012777  014000  151770          MOV     #FMT22!ECI,@RHOF ;16 BITS PER WORD
    113                                                                 ;ECC CORRECTION INHIBIT BECAUSE
    114                                                                 ;ECC IS NOT CHECKED HERE
    115 027260  005077  151766                  CLR     @RHCA           ;CYLINDER 0
    116 027264  004737  041514                  JSR     PC,CHECKT       ;CHECK THAT DVA,RDY,DPR,DRY = 1
        027270  104401  062726                  TYPE    ,CPHALT         ;AND THAT NO OTHERS = 1.  CANNOT CON-
                                                                        ;TINUE TESTING IF BOTH AREN'T TRUE
        027274  000000                          HALT                    ;STOP THE TEST
    117 027276  005037  001406                  CLR     ERFLG$  ;CLEAR ERROR FLAG
    118 027302  004737  046146                  JSR     PC,COMHD        ;READ DATA
    119
    120                             ;IF THE PROGRAM COMES BACK HERE WITHOUT ERROR PRINTOUT
    121                             ;FROM ''COMHD'' ROUTINE IN MEANS DATA IS TO BE CHECKED
    122
    123                             ;NOW THE DATA READ INTO ''REINTO'' BUFFER WILL
    124                             ;BE CHECKED, ONLY 10 WORDS SHOULD BE CHANGED
    125                             ;ALL OTHER WORDS SHOULD REMAIN UNCHANGED
    126                             ;THE ''WRFROM'' BUFFER IS FILLED WITH EXPECTED DATA AND CHECKED
    127
    128 027306  005737  001406                  TST     ERFLG$  ;HAVE ANY ERRORS OCCURED?
    129 027312  001053                          BNE     TST67           ;;BRANCH IF YES
    130 027314  004037  041376                  JSR     R0,CLAREA       ;CLEAR BUFFER
    131 027320  001510                          WRFROM                  ;FROM
    132 027322  002506                          WRFROM+776              ;TO
    133 027324  000000                          0                       ;DATA
    134
    135 027326  004037  041376                  JSR     R0,CLAREA       ;FILL EXPECTED DATA
    136 027332  001510                          WRFROM                  ;FROM
    137 027334  001532                          WRFROM+22               ;TO
    138 027336  177400                          177400                  ;DATA
    139
    140                             ;NOW READ DATA BUFFER IS CHECKED
    141
    142 027340  012700  001510                  MOV     #WRFROM,R0      ;GOOD DATA
    143 027344  012701  002554                  MOV     #REINTO,R1      ;DATA READ
    144 027350  012702  000400                  MOV     #256.,R2        ;COUNTER
    145 027354  012737  000401  046426  1$:     MOV     #257.,ERWORD    ;FOR ERROR WORD NO
    146 027362  022021                          CMP     (R0)+,(R1)+     ;COMPARE GOOD WITH READ BUFFER
    147 027364  001424                          BEQ     2$              ;BRANCH IF GOOD
    148 027366  014037  001124                  MOV     -(R0),$GDDAT    ;GOOD DATA
    149 027372  014137  001126                  MOV     -(R1),$BDDAT    ;BAD DATA
    150 027376  160237  046426                  SUB     R2,ERWORD       ;ERROR WORD NO
    151 027402  005737  001406                  TST     ERFLG$  ;ANY ERRORS ALREADY THERE
    152 027406  001002                          BNE     3$              ;IF YES BRANCH DO NOT TYPE HEADER
    153 027410  104004                          EMT     4
    154 027412  000401                          BR      4$              ;BRANCH TO AVOID PRINTING NEXT ERROR
    155 027414                      3$:
        027414  104005                          EMT     5
    156                                                                 ;WORDS
    157                                                                 ;WORD NUS 11-256 HAVE NOT BEEN
    158                                                                 ;READ AND BUFFER SHOULD BE
```

```
                                                        ;ZERO IF OTHER THAN ZERO
                                                        ;WRONG NUMBER OF WORDS HAVE
                                                        ;BEEN READ IN THE DISK NOW
                                                        ;CONTAINS 177400 ALL 256
                                                        ;WORDS BUT ONLY 10 WORDS
                                                        ;SHOULD BE READ IN

027416  022021                    4$:     CMP     (R0)+,(R1)+     ;UNDO -(R0) AND -(R1) FOR ERROR
027420  017746  151514                    MOV     @SWR,-(SP)      ;GET SWITCH SETTING
027424  042716  177177                    BIC     #177177,(SP)    ;KEEP ONLY SWITCH 7 AND 8
027430  022726  000200                    CMP     #SW07,(SP)+     ;IS 7 SET AND 8 RESET
027434  001402                            BEQ     TST67           ;BRANCH OUT IF YES
027436  005302                    2$:     DEC     R2              ;COUNT
027440  001345                            BNE     1$              ;BRANCH IF NOT COMPLETE

                                ;;*******************************************************************
                                ;*TEST 67       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
                                ;;*******************************************************************
027442  000004                    TST67:  SCOPE

                                ;DATA TABLE
                                ;20 WORDS OF 070707
                                ;THEN 16 WORDS WITH ZERO FLOATING FROM RIGHT
                                ;TO LEFT (EG. 177776,177775,177773 ETC)

027444  012706  001100                    MOV     #STACK,SP       ;RESET STACK
027450  012737  000067  001432            MOV     #67,TSTNM       ;MOVE #67 TO TEST NUMBER

                                ;SET UP "REINTO" FOR WHAT IS TO BE READ
027456  012701  002554                    MOV     #REINTO,R1      ;STARTING ADDRESS
027462  012721  010000                    MOV     #FMT22,(R1)+    ;CYLINDER 0 FORMAT 16 BIT WORDS
027466  012721  000401                    MOV     #401,(R1)+      ;TRACK=1, SECTOR=1
027472  005021                            CLR     (R1)+           ;KEY1=0
027474  005021                            CLR     (R1)+           ;KEY2=0
027476  004037  041376                    JSR     R0,CLAREA       ;FILL "REINTO" BUFFER
027502  002564                            .WORD   REINTO+<4*2>    ;FROM
027504  002632                            .WORD   REINTO+<23.*2>  ;TO
027506  070707                            .WORD   070707          ;DATA

027510  012700  177776                    MOV     #177776,R0      ;GETTING READY TO FLOAT 0
027514  012701  002634                    MOV     #REINTO+<24.*2>,R1;STARTING ADDRESS WHERE 177776 GOES
027520  010021                    1$:     MOV     R0,(R1)+        ;MOVE IN FLOATING 0
027522  000261                            SEC                     ;SET CARRY
027524  006100                            ROL     R0              ;GET 0 ONE BIT LEFT
027526  103774                            BCS     1$              ;BRANCH IF 16 NOT DONE

027530  004037  041376                    JSR     R0,CLAREA       ;FILL THE REST OF BUFFER WITH 0
027534  002674                            .WORD   REINTO+<40.*2>  ;FROM
027536  003552                            .WORD   REINTO+776      ;TO
027540  000000                            .WORD   0               ;DATA

                                ;SET UP SIMULATED DISK WITH WHAT IS TO BE READ
```

```
                                                    ;FILL "DISK" BUFFER
027542  004037  041376          JSR     R0,CLAREA
027546  050224                  .WORD   DISK
027550  050272                  .WORD   DISK+<19.*2>
027552  070707                  .WORD   070707
                                                    ;GETTING READY TO FLOAT ZEROS
027554  012700  177776          MOV     #177776,R0
027560  012701  050274          MOV     #DISK+<20.*2>,R1;STARTING ADDRESS WHERE 177776 GOES
027564  010021          2$:     MOV     R0,(R1)+        ;MOVE IN FLOATING 0
027566  000261                  SEC                     ;SET CARRY
027570  006100                  ROL     R0              ;GET 0 ONE BIT LEFT
027572  103774                  BCS     2$              ;BRANCH IF 16 NOT DONE
                                                    ;FILL THE REST OF BUFFER WITH 177777
027574  004037  041376          JSR     R0,CLAREA
027600  050334                  .WORD   DISK+<36.*2>    ;FROM
027602  051222                  .WORD   DISK+776        ;TO
027604  177777                  .WORD   177777          ;DATA
                                                    ;WRITE CHECK HEADER AND DATA
027606  004737  042204          JSR     PC,WRCHHD       ;CYLINDER 0, TRACK 1, SECTOR 1


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

                                                    ;GET UNIT NUMBER
027612  013746  001374          MOV     UNIT,-(SP)
027616  052716  000100          BIS     #IR,(SP)        ;ONLY BIT 6 SHOULD BE SET
027622  004737  041064          JSR     PC,PUTREG       ;SAVE REGISTERS
027626  022637  001312          CMP     (SP)+,CS2       ;COMPARE RHCS2
027632  001406                  BEQ     4$              ;BRANCH IF GOOD
027634  032712  040000          BIT     #WCE,@R2        ;WRITE CHECK ERROR HIGH?
027640  001402                  BEQ     3$              ;BRANCH IF ERROR NOT DUE TO "WCE"
027642  104017                  EMT     17
027644  000401                  BR      4$              ;RHBA CONTAINS ADDRESS+2
                                                    ;OF THE WORD IN MEMORY FROM
                                                    ;THE DISK THAT DID NOT COMPARE
                                                    ;TRE AND SC WILL BE SET DUE TO
                                                    ;WCE
027646          3$:
027646  104017                  EMT     17
                                                    ;BITS OTHER THAN IR
                                                    ;AND UNIT NO. WAS SET


                        ;NOW CHECK MEMORY TO SEE IF NOTHING GOT DESTROYED
                        ;FILL "WRFROM" WITH WHAT SHOULD BE IN "REINTO" THEN CHECK

027650  012700  001510  4$:     MOV     #WRFROM,R0      ;STARTING ADDRESS
027654  012720  010000          MOV     #FMT22,(R0)+    ;CYLINDER
027660  012720  000401          MOV     #401,(R0)+      ;TRACK=1, SECTOR=1
027664  005020                  CLR     (R0)+           ;KEY1=0
027666  005020                  CLR     (R0)+           ;KEY2=0

027670  004037  041376          JSR     R0,CLAREA       ;FILL "WRFROM" BUFFER
027674  001520                  .WORD   WRFROM+<4*2>    ;FROM
027676  001566                  .WORD   WRFROM+<23.*2>  ;TO
027700  070707                  .WORD   070707          ;DATA

027702  012700  177776          MOV     #177776,R0      ;GETTING READY TO FLOAT 0
```

```
                                                ;STARTING ADDRESS WHERE 177776 GOES
                                                ;MOVE IN FLOATING 0
                                                ;SET CARRY
                                                ;GET 0 ONE BIT LEFT
                                                ;BRANCH IF 16 NOT DONE

                                                ;FILL THE REST OF BUFFER WITH 0
                                                ;FROM
                                                ;TO
                                                ;DATA

                                                ;CLEAR ERROR FLAG

                                                ;CHECK
                                                ;GOOD BUFFER
                                                ;TEST BUFFER
                                                ;NUMBER OF WORDS CHECKED
                                                ;RETURN POINT FOR ERROR HEADER
                                                ;RETURN POINT FOR ERROR DATA
                                                ;RETURN FOR GOOD COMPARISON

                                                ;RETURN TO COMPARISON SUBROUTINE

                                                ;CHANGED AFTER A WRITE
                                                ;CHECK HEADER AND DATA COMMAND
                                                ;WORD NO CONTAINS THE WORD
                                                ;NUMBER THAT GOT CHANGED
                                                ;RETURN TO COMPARISON SUBROUTINE

                                                ;ONLY A BRANCH POINT

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

                                                ;RESET STACK
                                                ;MOVE #70 TO TEST NUMBER

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

                                                ;GETTING READY TO FLOAT 1
                                                ;STARTING ADDRESS WHERE 1 GOES
                                                ;MOVE FLOATING 1
```

```
                                                                    ;GET 1 ONE BIT LEFT
                                                                    ;BRANCH IF 16 NOT DONE
                                                                    ;GETTING READY TO FLOAT 0
                                                                    ;STARTING ADDRESS WHERE 177776 GOES
                                                                    ;MOVE IN FLOATING 0
                                                                    ;SET CARRY
                                                                    ;GET 0 ONE BIT LEFT
                                                                    ;BRANCH IF 16 NOT DONE

                                                                    ;FILL REST OF BUFFER WITH 1
                                                                    ;FROM
                                                                    ;TO
                                                                    ;WITH DATA

                                                                    ;SET UP SIMULATED DISK WITH WHAT IS TO BE READ

                                                                    ;GETTING READY TO FLOAT 1
                                                                    ;STARTING ADDRESS WHERE 1 GOES
                                                                    ;MOVE FLOATING 1
                                                                    ;GET 1 ONE BIT LEFT
                                                                    ;BRANCH IF 16 NOT DONE

                                                                    ;GETTING READY TO FLOAT 0
                                                                    ;STARTING ADDRESS WHERE 177776 GOES
                                                                    ;MOVE FLOATING 0
                                                                    ;SET CARRY
                                                                    ;GET 0 ONE BIT LEFT
                                                                    ;BRANCH IF 16 NOT DONE

                                                                    ;FILL REST OF BUFFER WITH 0
                                                                    ;FROM
                                                                    ;TO
                                                                    ;WITH DATA

                                                                    ;WRITE CHECK DATA
                                                                    ;CYLINDER 0, TRACK 1, SECTOR 1
                                                                    ;KEYS 0, 32 WORDS.

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

                                                                    ;GET UNIT NUMBER
                                                                    ;ONLY BIT 6 SHOULD BE SET
                                                                    ;SAVE REGISTERS
                                                                    ;COMPARE RHCS2
                                                                    ;BRANCH IF GOOD
                                                                    ;WRITE CHECK ERROR HIGH?
                                                                    ;BRANCH IF ERROR NOT DUE TO 'WCE'

                                                                    ;RHBA CONTAINS ADDRESS+2
                                                                    ;OF THE WORD IN MEMORY FROM
                                                                    ;THE DISK THAT DID NOT COMPARE
                                                                    ;TRE AND SC WILL BE SET DUE TO WCE

                                                                    ;BUT SOME BITS OTHER THAN
                                                                    ;IR AND UNIT NO. WERE SET
```

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

030172  005037  001406    6$:     CLR     ERFLG$  ;CLEAR ERROR FLAG
030176  012700  000001            MOV     #1,R0           ;GETTING READY TO FLOAT 1
030202  012701  001510            MOV     #WRFROM,R1      ;START ADDRESS WHERE 1 GOES
030206  010021            7$:     MOV     R0,(R1)+        ;MOVE FLOATING 1
030210  006100                    ROL     R0              ;GET 1 ONE BIT LEFT
030212  103375                    BCC     7$              ;BRANCH IF 16 NOT DONE

030214  012700  177776            MOV     #177776,R0      ;GETTING READY TO FLOAT 0
030220  012701  001550            MOV     #WRFROM+<16.*2>,R1 ;STARTING ADDRESS WHERE 177776 GOES
030224  010021            10$:    MOV     R0,(R1)+        ;MOVE IN FLOATING 0
030226  000261                    SEC                     ;SET CARRY
030230  006100                    ROL     R0              ;GET 0 ONE BIT LEFT
030232  103774                    BCS     10$             ;BRANCH IF CARRY SET

030234  004037  041376            JSR     R0,CLAREA       ;FILL REST OF BUFFER WITH 1
030240  001610                    .WORD   WRFROM+<32.*2>  ;FROM
030242  002506                    .WORD   WRFROM+776      ;TO
030244  000001                    .WORD   1               ;WITH DATA

                                ;NOW THE READ BUFFER WILL BE CHECKED

030246  004037  042354            JSR     R0,COMPAR       ;CHECK
030252  001510                    WRFROM                  ;GOOD BUFFER
030254  002554                    REINTO                  ;TEST BUFFER
030256  000400                    256.                    ;NUMBER OF WORDS CHECKED
030260  030266                    11$                     ;RETURN POINT FOR ERROR HEADER
030262  030272                    12$                     ;RETURN POINT FOR ERROR DATA

030264  030300                    TST71                   ;RETURN FOR GOOD COMPARISON

030266                  11$:
030266  104004                    EMT     4
030270  000207                    RTS     PC              ;RETURN TO COMPARISON SUBROUTINE
030272                  12$:
030272  104005                    EMT     5
                                                        ;CHANGED AFTER A WRITE
                                                        ;CHECK DATA COMMAND
                                                        ;WORD NO CONTAINS THE WORD
                                                        ;NUMBER THAT GOT CHANGED
030274  000207                    RTS     PC              ;RETURN TO COMPARISON SUBROUTINE

030276  000240          13$:      NOP                     ;ONLY A BRANCH POINT
```

```
                                .SBTTL  ERROR BIT FUNCTIONAL TESTS
                                ;;*********************************************************************
                                ;*TEST 71        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
                                ;;*********************************************************************
030300  000004                  TST71:  SCOPE
030302  012706  001100                  MOV     #STACK,SP       ;RESET STACK
030306  012737  000071  001432          MOV     #71,TSTNM       ;MOVE #71 TO TEST NUMBER
030314  004737  041460                  JSR     PC,CLDISK       ;CLEAR DISK REGISTERS
030320  004737  041514                  JSR     PC,CHECKT       ;CHECK THAT DVA,RDY,DPR,DRY = 1
030324  104401  062726                  TYPE    ,CPHALT         ;AND THAT NO OTHERS = 1.  CANNOT CON-
                                                                ;TINUE TESTING IF BOTH AREN'T TRUE
030330  000000                          HALT                    ;STOP THE TEST

030332  012700  002554                  MOV     #REINTO,R0      ;BUFFER STARTING FOR 3 ERROR
                                                                ;REGISTERS
030336  013720  001242                  MOV     RHER1,(R0)+     ;RHER1 STORED IN REINTO
030342  012720  000000                  MOV     #0,(R0)+        ;BITS NOT TO BE CHECKED IN RHER1
030346  013720  001246                  MOV     RHER2,(R0)+     ;RHER2 STORED IN REINTO+4
030352  012720  000000                  MOV     #0,(R0)+        ;BITS NOT TO BE CHECKED IN RHER2
030356  013720  001254                  MOV     RHER3,(R0)+     ;RHER3 STORED IN REINTO+10
030362  012720  000000                  MOV     #0,(R0)+        ;BITS NOT TO BE CHECKED IN RHER3

030366  013704  001256                  MOV     RHAS,R4 ;R4 HAS RHAS
030372  013705  001416                  MOV     ATTENT,R5       ;R5 HAS ATA BIT IN RHAS
030376  012737  030424  001110          MOV     #2$,$LPERR      ;THAT SHOULD SET WITH ERROR
                                                                ;RETURN POINT TO ERROR
030404  012737  000003  001200          MOV     #3,$TMP1        ;ERROR REGISTER COUNTER
030412  012700  002554                  MOV     #REINTO,R0      ;REGISTER BUFFER POINTER

030416  012002                  1$:     MOV     (R0)+,R2        ;R2 HAS ADDRESS OF ERROR REG
030420  012701  000001                  MOV     #BIT0,R1        ;R1 WILL HAVE BIT UNDER TEST
030424  052777  000040  150604  2$:     BIS     #CLR,@RHCS2     ;CLEAR RHCS2
030432  013777  001374  150576          MOV     UNIT,@RHCS2     ;REINSTATE UNIT NO.
030440  010112                          MOV     R1,@R2          ;SET ERROR BIT
030442  004737  041064                  JSR     PC,PUTREG       ;READ AND SAVE REGISTERS
030446  120537  001332                  CMPB    R5,AS           ;ONLY THE BIT IN R5 SHOULD BE
                                                                ;SET IN RHAS
030452  001401                          BEQ     3$              ;LOOK @ RHDS1 IF GOOD
030454  104020                          EMT     20

                                                                ;ERROR BIT IN AN ERROR
                                                                ;REGISTER, THE CORRESPONDING
                                                                ;RHAS BIT DID NOT SET

030456  013746  001336          3$:     MOV     DS1,-(SP)       ;GET RHDS1
030462  042716  001100                  BIC     #VV!PROG,(SP)   ;REMOVE VV AND PROG
```

```
     030466  022726  140600                  CMP     #ATA!ERR!DPR!DRY,(SP)+;THESE BITS PLUS VV SHOULD BE IN RHDS1
     030472  001401                          BEQ     4$              ;CHECK 'GO' NEXT, IF THIS WAS OK
     030474  104020                          EMT     20
                                                                     ;ERROR BIT, COMPOSITE ERROR
                                                                     ;OR ATTENTION ACTIVE, OR
                                                                     ;ONE OF THE OTHER
                                                                     ;PERMANENT BITS DID NOT SET
     030476  012777  000001  150534  4$:     MOV     #GO,@RHCS1      ;GIVE NO-OP
     030504  004737  041064                  JSR     PC,PUTREG       ;SAVE REGISTERS
     030510  020112                          CMP     R1,@R2          ;GO SHOULD NOT CLEAR ERROR
     030512  001410                          BEQ     5$              ;FURTHER CHECK OF 'GO' FUNCTIONALITY
     030514  010237  041130                  MOV     R2,REGADR       ;FAILING REGISTER
     030520  010137  001124                  MOV     R1,$GDDAT       ;GOOD DATA
     030524  013737  001312  001126          MOV     CS2,$BDDAT      ;BAD DATA
     030532  104001                          EMT     1
                                                                     ;ERROR REGISTER
     030534  013746  001336          5$:     MOV     DS1,-(SP)       ;GET RHDS1
     030540  042716  001100                  BIC     #VV!PROG,(SP)   ;CLEAR VV AND PROG
     030544  022726  140600                  CMP     #ATA!ERR!DPR!DRY,(SP)+;GO SHOULD NOT CLEAR ANY BITS
     030550  001401                          BEQ     7$              ;CHECK NEXT ERROR BIT IF A-OK
     030552  104020                          EMT     20
                                                                     ;ATA AND/OR ERR

                                     ;THIS IS THE MAIN BIT TESTING CONTROL LOGIC

     030554  006301                  7$:     ASL     R1              ;GET NEXT BIT TO THE LEFT
     030556  103403                          BCS     10$             ;GO ON TO NEXT REGISTER IF DONE
     030560  031001                          BIT     (R0),R1         ;IS THIS BIT TO BE TESTED ?
     030562  001374                          BNE     7$              ;IF NOT, GET NEXT ONE
     030564  000717                          BR      2$              ;IF TO BE TESTED, GO DO IT !

     030566  005720                  10$:    TST     (R0)+           ;ADVANCE R0 TO NEXT ERROR REG.
     030570  005337  001200                  DEC     $TMP1           ;REGISTER COUNTER
     030574  001310                          BNE     1$              ;DO NEXT ONE, IF 3 NOT COMPLETE


                                     ;NOW AFTER SETTING ATA IN RHDS1 "1" IN RHAS AT THE
                                     ;DRIVE POSITION SHOULD CLEAR ATA IN RHDS1

     030576  004737  041460          11$:    JSR     PC,CLDISK       ;CLEAR
     030602  012737  030576  001110          MOV     #11$,$LPERR     ;ERROR RETURN
     030610  012714  177777                  MOV     #-1,@R4         ;SET BIT IN RHAS AND ATA IN RHDS1
     030614  013777  001416  150434          MOV     ATTENT,@RHAS    ;WRITE 1 INTO  DRIVE BIT POSITION
     030622  004737  041064                  JSR     PC,PUTREG       ;SAVE REGISTERS
     030626  105737  001332                  TSTB    AS              ;THIS SHOULD BE ZERO
     030632  001401                          BEQ     12$
     030634  104020                          EMT     20
                                                                     ;AT THE DRIVE BIT POSITION
                                                                     ;DID NOT CLEAR IT

     030636  013746  001336          12$:    MOV     DS1,-(SP)       ;GET RHDS1
     030642  042716  001000                  BIC     #PROG,(SP)      ;MASK PROGRAMABLE
     030646  022726  040700                  CMP     #ERR!VV!DPR!DRY,(SP)+
                                                                     ;RHDS1 SHOULD HAVE THESE BITS
                                                                     ;BUT ATA SHOULD BE CLEARED
```

```
                                        BEQ     13$             ;CHECK RHER1 IF GOOD
                                        EMT     20
                                                                ;DRIVE BIT POSITION DID NOT
                                                                ;CLEAR ATA IN RHDS1


                        177777  001316  13$:    CMP     #-1,ER1 ;RHER1 SHOULD NOT CHANGE
                                                                ;BY CLEARING RHAS
                                        BEQ     TST72           ;BRANCH IF GOOD
                                        EMT     20
                                                                ;RHAS BY MOVING "1" INTO
                                                                ;THE DRIVE BIT POSITION


                                ;;*****************************************************************
                                ;*TEST 72       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)
                                ;;*****************************************************************
000004                          TST72:  SCOPE
                        001100          MOV     #STACK,SP       ;RESET STACK

                        000072  001432  MOV     #72,TSTNM       ;MOVE #72 TO TEST NUMBER
                        041376          JSR     R0,CLAREA       ;CLEAR SIMULATED DISK
                                        .WORD   DISK            ;FROM
                                        .WORD   DISK+776        ;TO
                                        .WORD   177400          ;DATA

                        041376          JSR     R0,CLAREA       ;CLEAR READ INTO BUFFER
                                        .WORD   REINTO          ;FROM
                                        .WORD   REINTO+776      ;TO
                                        .WORD   0               ;DATA

                                ;THESE ARE TO SETUP FOR DISKLESS USE ONLY

                        010000  046306  MOV     #FMT22,CYL      ;CYLINDER 0 16 BITS PER WORD FORMAT
                        046311          CLRB    SECOTR+1        ;TRACK 0
                        000001  046310  MOVB    #1,SECOTR       ;SECTOR 1
                        046312          CLR     KEY1            ;KEY1=0
                        046314          CLR     KEY2            ;KEY2=0
                        000012  046366  MOV     #10.,DAWORD     ;NO. OF DATA WORDS
                        046316          CLR     X               ;THIS IS A READ COMMAND
                        042666          JSR     R5,CRC  ;GO TO CALCULATE CRC
                        046306          CYL
                        050206          WCRC

                                ;THESE ARE REGULAR SETUPS

                        041460          JSR     PC,CLDISK       ;SETUP GENERAL REGISTERS
                        001466          MOV     READAT,@R1      ;READ DATA INTO RHCS1=70
                        177766  150212  MOV     #-10.,@RHWC     ;10 DATA WORDS
                        002554  150206  MOV     #REINTO,@RHBA   ;STARTING ADDRESS OF READ BUFFER
                        000001          MOVB    #1,-(SP)        ;IN LOWER BYTE GET SECTOR 1
                        000000  000001  MOVB    #0,1(SP)        ;GET TRACK0 IN UPPER BYTE
                        150200          MOV     (SP)+,@RHDST    ;TRACK/SECTOR IN RHDST
                        014000  150176  MOV     #FMT22!ECI,@RHOF ;16 BITS PER WORD
```

```
                                                        ;ECC CORRECTION INHIBIT BECAUSE
                                                        ;ECC IS NOT CHECKED HERE
031052  005077  150174                CLR     @RHCA       ;CYLINDER 0
031056  004737  041514                JSR     PC,CHECKT   ;CHECK THAT DVA,RDY,DPR,DRY = 1
031062  104401  062726                TYPE    ,CPHALT     ;AND THAT NO OTHERS = 1.  CANNOT CON-
                                                        ;TINUE TESTING IF BOTH AREN'T TRUE
031066  000000                        HALT                ;STOP THE TEST
031070  052777  000010  150140        BIS     #BAI,@RHCS2 ;SET BUS ADDRESS INHIBIT
031076  005037  001406                CLR     ERFLG$  ;CLEAR ERROR FLAG
031102  004737  046146                JSR     PC,COMHD    ;READ DATA

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

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

031106  005037  001406                CLR     ERFLG$  ;CLEAR FLAG
031112  004037  041376                JSR     R0,CLAREA   ;CLEAR BUFFER
031116  001510                        WRFROM              ;FROM
031120  002506                        WRFROM+776          ;TO
031122  000000                        0                   ;DATA

                                    ;EXPECTED DATA IS 177400 IN FIRST LOCATION ONLY
031124  012737  177400  001510        MOV     #177400,WRFROM  ;EXPECTED DATA

                                    ;NOW READ DATA BUFFER IS CHECKED

031132  012700  001510                MOV     #WRFROM,R0  ;GOOD DATA
031136  012701  002554                MOV     #REINTO,R1  ;DATA READ
031142  012702  000400                MOV     #256.,R2    ;COUNTER
031146  012737  000401  046426  1$:   MOV     #257.,ERWORD    ;FOR ERROR WORD NO
031154  022021                        CMP     (R0)+,(R1)+ ;COMPARE GOOD WITH READ BUFFER
031156  001424                        BEQ     2$          ;BRANCH IF GOOD
031160  014037  001124                MOV     -(R0),$GDDAT    ;GOOD DATA
031164  014137  001126                MOV     -(R1),$BDDAT    ;BAD DATA
031170  160237  046426                SUB     R2,ERWORD   ;ERROR WORD NO
031174  005737  001406                TST     ERFLG$      ;ANY ERRORS ALREADY THERE
031200  001002                        BNE     3$          ;IF YES BRANCH DO NOT TYPE HEADER
031202  104004                        EMT     4
031204  000401                        BR      4$          ;BRANCH TO AVOID PRINTING NEXT ERROR
031206                          3$:
031206  104005                        EMT     5
                                                        ;WORDS
                                                        ;WORD NOS 11-256 HAVE NOT BEEN
                                                        ;READ AND BUFFER SHOULD BE
                                                        ;ZERO IF OTHER THAN ZERO
                                                        ;WRONG NUMBER OF WORDS HAVE
                                                        ;BEEN READ IN THE DISK NOW
                                                        ;CONTAINS 177400 ALL 256
                                                        ;WORDS BUT ONLY 10 WORDS
                                                        ;SHOULD BE READ IN

031210  022021                4$:     CMP     (R0)+,(R1)+ ;UNDO -(R0) AND -(R1) FOR ERROR
031212  017746  147722                MOV     @SWR,-(SP)  ;GET SWITCH SETTING
```

```
                                        BIC     #177177,(SP)    ;KEEP ONLY SWITCH 7 AND 8
                                        CMP     #SW07,(SP)+     ;IS 7 SET AND 8 RESET
                                        BEQ     TST73           ;BRANCH OUT IF YES
                                2$:     DEC     R2              ;COUNT
                                        BNE     1$              ;BRANCH IF NOT COMPLETE

                                ;;*****************************************************************
                                ;*TEST 73       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
                                ;;*****************************************************************
031234  000004                  TST73:  SCOPE
031236  012706  001100                  MOV     #STACK,SP       ;RESET STACK
031242  012737  000073  001432          MOV     #73,TSTNM       ;MOVE #73 TO TEST NUMBER

                                ;CHECK TO SEE IF THE PROGRAM IS RUNNING WITH AN RH70

031250  005737  001436                  TST     RH70            ;TEST FLAG FOR RH70 CONTROLLER
031254  001402                          BEQ     64$             ;IF FLAG = 1, THIS TEST IS SKIPPED
031256  000137  031600                  JMP     TST74           ;JUMP TO NEXT TEST
031262                          64$:                            ;IF FLAG = 0, DO THIS TEST

031262  004037  041376                  JSR     R0,CLAREA       ;CLEAR SIMULATED DISK
031266  050224                          .WORD   DISK            ;FROM
031270  051222                          .WORD   DISK+776        ;TO
031272  177400                          .WORD   177400          ;DATA


                                ;THESE ARE TO SETUP FOR DISKLESS USE ONLY

031274  012737  010000  046306          MOV     #FMT22,CYL      ;CYLINDER 0, 16 BITS PER WORD FORMAT
031302  105037  046311                  CLRB    SECOTR+1        ;TRACK 0
031306  112737  000001  046310          MOVB    #1,SECOTR       ;SECTOR 1
031314  005037  046312                  CLR     KEY1            ;KEY1=0
031320  005037  046314                  CLR     KEY2            ;KEY2=0
031324  012737  000001  046366          MOV     #1.,DAWORD      ;NO. OF DATA WORDS
031332  005037  046316                  CLR     X               ;THIS IS A READ COMMAND
031336  004537  042666                  JSR     R5,CRC  ;GO TO CALCULATE CRC
031342  046306                          CYL
031344  050206                          WCRC

                                ;THESE ARE REGULAR SETUPS

031346  004737  041460                  JSR     PC,CLDISK       ;SETUP GENERAL REGISTERS
031352  013711  001466                  MOV     READAT,@R1      ;READ DATA INTO RHCS1=70
031356  012777  177777  147646          MOV     #-1.,@RHWC      ;10 DATA WORDS
031364  012777  160000  147642          MOV     #160000,@RHBA   ;STARTING ADDRESS OF READ BUFFER
031372  052711  001400                  BIS     #A16!A17,@R1    ;IS 760000
031376  112746  000001                  MOVB    #1,-(SP)        ;IN LOWER BYTE GET SECTOR 1
031402  112766  000000  000001          MOVB    #0,1(SP)        ;GET TRACK0 IN UPPER BYTE
031410  012677  147630                  MOV     (SP)+,@RHDST    ;TRACK/SECTOR IN RHDST
031414  012777  014000  147626          MOV     #FMT22!ECI,@RHOF ;16 BITS PER WORD
                                                                ;ECC CORRECTION INHIBIT BECAUSE
                                                                ;ECC IS NOT CHECKED HERE
```

```
    270 031422  005077  147624                CLR     @RHCA           ;CYLINDER 0
    271 031426  004737  041514                JSR     PC,CHECKT       ;CHECK THAT DVA,RDY,DPR,DRY = 1
        031432  104401  062726                TYPE    ,CPHALT         ;AND THAT NO OTHERS = 1.  CANNOT CON-
                                                                      ;TINUE TESTING IF BOTH AREN'T TRUE
        031436  000000                        HALT                    ;STOP THE TEST
    272 031440  052777  000010  147570        BIS     #BAI,@RHCS2     ;SET BUS ADDRESS INHIBIT
    273 031446  005037  001406                CLR     ERFLG$  ;CLEAR ERROR FLAG
    274 031452  004737  046146                JSR     PC,COMHD        ;READ DATA
    275
    276
    277
    278 031456  011137  001126          1$:   MOV     @R1,$BDDAT      ;TEST DATA
    279
    280 031462  022737  145670  001126        CMP     #SC!TRE!DVA!A16!A17!RDY!70,$BDDAT ;COMPARE RHCS1
    281 031470  001406                        BEQ     2$              ;BRANCH IF GOOD
    282 031472  012737  144270  001124        MOV     #SC!TRE!DVA!RDY!70,$GDDAT ;GOOD DATA
    283 031500  010137  041130                MOV     R1,REGADR       ;REGISTER RHCS1
    284 031504  104001                        EMT     1
    285                                                               ;MEMORY DID NOT SET
    286                                                               ;REQUIRED BITS
    287 031506  013746  001374          2$:   MOV     UNIT,-(SP)      ;GET UNIT NUMBER
    288 031512  052716  004110                BIS     #NEM!IR!BAI,(SP)        ;INCLUDE NEM BAI AND IR
    289 031516  012637  001124                MOV     (SP)+,$GDDAT    ;
    290 031522  011237  001126                MOV     @R2,$BDDAT      ;TEST DATA
    291 031526  023737  001124  001126        CMP     $GDDAT,$BDDAT;COMPARE RHCS2
    292 031534  001403                        BEQ     3$
    293 031536  010237  041130                MOV     R2,REGADR       ;REGISTER ADDRESS
    294 031542  104001                        EMT     1
    295                                                               ;CAUSED AN ERROR SHOULD SET NEM
    296 031544  017737  147464  001126  3$:   MOV     @RHBA,$BDDAT    ;TEST DATA
    297
    298 031552  022737  160000  001126        CMP     #160000,$BDDAT  ;COMPARE RHBA
    299 031560  001407                        BEQ     4$              ;BRANCH IF GOOD
    300 031562  012737  160000  001124        MOV     #160000,$GDDAT;GOOD DATA
    301 031570  013737  001234  041130        MOV     RHBA,REGADR     ;REGISTER ADDRESS RHBA
    302 031576  104001                        EMT     1
    303                                                               ;RHBA DOES NOT HAVE 160002
    304 031600                          4$:
    305
    315                                       ;;*****************************************************************
                                              ;*TEST 74       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
                                              ;;*****************************************************************
        031600  000004                        TST74:  SCOPE
    316
    317                                       ;DATA TABLE
    318                                       ;TOTAL OF 32 WORDS CONSISTING OF
    319                                       ;16 WORDS OF FLOATING ONES (EG. 1, 2, 4, 10)
    320                                       ;16 WORDS OF FLOATING ZEROS (EG. 177776, 177775)
    321
    322 031602  012706  001100                      MOV     #STACK,SP       ;RESET STACK
```

```
031606  012737  000074  001432          MOV     #74,TSTNM       ;MOVE #74 TO TEST NUMBER
031614  004737  041460                  JSR     PC,CLDISK       ;INIT AND SET UP GENERAL REGISTERS

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

031620  012700  000001                  MOV     #1,R0           ;GETTING READY TO FLOAT 1
031624  012701  002554                  MOV     #REINTO,R1      ;STARTING ADDRESS WHERE 1 GOES
031630  010021                  1$:     MOV     R0,(R1)+        ;MOVE FLOATING 1
031632  006100                          ROL     R0              ;GET 1 ONE BIT LEFT
031634  103375                          BCC     1$              ;BRANCH IF 16 NOT DONE
031636  012700  177776                  MOV     #177776,R0      ;GETTING READY TO FLOAT 0
031642  012701  002614                  MOV     #REINTO+<16.*2>,R1 ;STARTING ADDRESS WHERE 177776 GOES
031646  010021                  2$:     MOV     R0,(R1)+        ;MOVE IN FLOATING 0
031650  000261                          SEC                     ;SET CARRY
031652  006100                          ROL     R0              ;GET 0 ONE BIT LEFT
031654  103774                          BCS     2$              ;BRANCH IF 16 NOT DONE

031656  004037  041376                  JSR     R0,CLAREA       ;FILL REST OF BUFFER WITH 1
031662  002654                          .WORD   REINTO+<32.*2>  ;FROM
031664  003552                          .WORD   REINTO+776      ;TO
031666  000001                          .WORD   1               ;WITH DATA

                                ;SET UP SIMULATED DISK WITH WHAT IS TO BE READ

031670  012700  000001                  MOV     #1,R0           ;GETTING READY TO FLOAT 1
031674  012701  050224                  MOV     #DISK,R1        ;STARTING ADDRESS WHERE 1 GOES
031700  010021                  3$:     MOV     R0,(R1)+        ;MOVE FLOATING 1
031702  006100                          ROL     R0              ;GET 1 ONE BIT LEFT
031704  103375                          BCC     3$              ;BRANCH IF 16 NOT DONE

031706  012700  177776                  MOV     #177776,R0      ;GETTING READY TO FLOAT 0
031712  012701  050264                  MOV     #DISK+<16.*2>,R1 ;STARTING ADDRESS WHERE 177776 GOES
031716  010021                  4$:     MOV     R0,(R1)+        ;MOVE FLOATING 0
031720  000261                          SEC                     ;SET CARRY
031722  006100                          ROL     R0              ;GET 0 ONE BIT LEFT
031724  103774                          BCS     4$              ;BRANCH IF 16 NOT DONE

031726  004037  041376                  JSR     R0,CLAREA       ;FILL REST OF BUFFER WITH 0
031732  050324                          .WORD   DISK+<32.*2>    ;FROM
031734  051222                          .WORD   DISK+776        ;TO
031736  000000                          .WORD   0               ;WITH DATA

                                ;CHANGE FIFTH WORD TO 0 ON DISK

031740  005037  050234                  CLR     DISK+10 ;CLEAR FIFTH WORD ON DISK
031744  005037  001406                  CLR     ERFLG$  ;CLEAR ERROR FLAG
031750  004737  042516                  JSR     PC,WRCHDA       ;WRITE CHECK DATA
                                                                ;CYLINDER 0, TRACK 1, SECTOR 1
                                                                ;KEYS 0, 32 WORDS.

                                ;IF THE PROGRAM COMES BACK HERE THEN WRITE CHECK
                                ;HAS BEEN COMPLETED, NOW WRITE CHECK ERROR BIT IS TESTED
                                ;ALONG WITH RHWC FOR PROPER WORD COUNT AND RHBA FOR ADDRESS

031754  013746  001374                  MOV     UNIT,-(SP)      ;GET UNIT NUMBER
031760  052716  040300                  BIS     #IR!OR!WCE,(SP) ;ONLY BIT 6 SHOULD BE SET
031764  004737  041064                  JSR     PC,PUTREG       ;SAVE REGISTERS
```

```
                                                CMP     (SP)+,CS2       ;COMPARE RHCS2
                                                BEQ     6$              ;BRANCH IF GOOD
                                                BIT     #WCE,CS2        ;WRITE CHECK ERROR HIGH?
                                                BNE     5$              ;BRANCH IF ERROR NOT DUE TO 'WCE'
                                                EMT     17
                                                BR      6$              ;RHBA CONTAINS ADDRESS+2
                                                                        ;OF THE WORD IN MEMORY FROM
                                                                        ;THE DISK THAT DID NOT COMPARE
                                                                        ;TRE AND SC WILL BE SET DUE TO WCE
                                        5$:
                                                EMT     17
                                                                        ;BUT SOME BITS OTHER THAN
                                                                        ;IR AND UNIT NO. WERE SET
                                        6$:     TST     RH70            ;TEST FOR RH70 CONTROLLER
                                                BEQ     16$             ;SKIP RH70 CODE AND DO RH11 IF NOT
                                                CMP     #-24.,WC        ;COMPARE RHWC AFTER A FORCED
                                                                        ;WRITE CHECK ERROR
                                                BEQ     17$             ;CHECK RHBA IF GOOD
                                                EMT     17
                                                                        ;FORCED WRITE CHECK ERROR ON FIFTH WORD
                                                BR      15$             ;BRANCH TO CONTINUE TEST
                                        17$:    CMP     #REINTO+<8.*2>,BA ;COMPARE RHBA AFTER A FORCED
                                                                        ;WRITE CHECK ERROR IN FIFTH WORD
                                                BEQ     15$             ;CONTINUE IF GOOD
                                                EMT     17
                                                                        ;FORCED WRITE CHECK ERROR ON FIFTH WORD
                                                BR      15$             ;SKIP RH11 CODE AND CONTINUE WITH TEST
                                        16$:    CMP     #-27.,WC        ;COMPARE RHWC AFTER A FORCED
                                                                        ;WRITE CHECK ERROR
                                                BEQ     14$             ;CHECK RHBA IF GOOD
                                                EMT     17
                                                                        ;FORCED WRITE CHECK ERROR ON FIFTH WORD
                                                BR      15$             ;BRANCH TO CONTINUE TEST
                                        14$:    CMP     #REINTO+<5.*2>,BA;COMPARE RHBA AFTER FORCED
                                                                        ;WRITE CHECK ERROR IN FIFTH WORD
                                                BEQ     15$             ;CONTINUE IF GOOD
                                                EMT     17
                                                                        ;FORCED WRITE CHECK ERROR ON FIFTH WORD

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

                                15$:    CLR     ERFLG$  ;CLEAR ERROR FLAG
                                                MOV     #1,R0           ;GETTING READY TO FLOAT 1
                                                MOV     #WRFROM,R1      ;START ADDRESS WHERE 1 GOES
                                        7$:     MOV     R0,(R1)+        ;MOVE FLOATING 1
                                                ROL     R0              ;GET 1 ONE BIT LEFT
                                                BCC     7$              ;BRANCH IF 16 NOT DONE

                                                MOV     #177776,R0      ;GETTING READY TO FLOAT 0
                                                MOV     #WRFROM+<16.*2>,R1 ;STARTING ADDRESS WHERE 177776 GOES
                                        10$:    MOV     R0,(R1)+        ;MOVE IN FLOATING 0
```

```
                                        SEC                     ;SET CARRY
                                        ROL     R0              ;GET 0 ONE BIT LEFT
                                        BCS     10$             ;BRANCH IF CARRY SET

                                        JSR     R0,CLAREA       ;FILL REST OF BUFFER WITH 1
                                        .WORD   WRFROM+<32.*2>  ;FROM
                                        .WORD   WRFROM+776      ;TO
                                        .WORD   1               ;WITH DATA

                                ;NOW THE READ BUFFER WILL BE CHECKED

                                        JSR     R0,COMPAR       ;CHECK
                                        WRFROM                  ;GOOD BUFFER
                                        REINTO                  ;TEST BUFFER
                                        256.                    ;NUMBER OF WORDS CHECKED
                                        11$                     ;RETURN POINT FOR ERROR HEADER
                                        12$                     ;RETURN POINT FOR ERROR DATA

                                        TST75                   ;RETURN FOR GOOD COMPARISON

                                11$:
                                        EMT     4
                                        RTS     PC              ;RETURN TO COMPARISON SUBROUTINE
                                12$:
                                        EMT     5
                                                                ;CHANGED AFTER A WRITE
                                                                ;CHECK DATA COMMAND
                                                                ;WORD NO CONTAINS THE WORD
                                                                ;NUMBER THAT GOT CHANGED
                                        RTS     PC              ;RETURN TO COMPARISON SUBROUTINE

                                13$:    NOP                     ;ONLY A BRANCH POINT

                                ;;*****************************************************************
                                ;*TEST 75        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=RIT4 SHOULD SET BUT THE
                                ;*READ SHOULD BE COMPLETE
                                ;;*****************************************************************
032206  000004         TST75:  SCOPE
010706  001100                 MOV     #STACK,SP       ;RESET STACK
012737  000075  001432         MOV     #75,TSTNM       ;MOVE #75 TO TEST NUMBER

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

032222  012746  125252         MOV     #125252,-(SP)   ;DATA TO BE READ
032226  012705  000400         MOV     #256.,R5        ;COUNTER
032232  012700  050224         MOV     #DISK,R0        ;START OF SIMULATED DISK DATA
032236  011620          1$:    MOV     (SP),(R0)+      ;MOVE IN DATA ON TO SIMULATED DISK
032240  005305                 DEC     R5              ;COUNT
032242  001375                 BNE     1$              ;BRANCH IF 256 NOT COMPLETE
032244  005726                 TST     (SP)+           ;UNDO -(SP)
032246  012705  000021         MOV     #17.,R5         ;2 ECC WORDS
                                                        ;1 DATA GAP
```

```
                                                        ;14 TOLERANCE GAP
032252  005020                  2$:     CLR     (R0)+   ;CLEAR ECC, DATA GAP, AND
032254  005305                          DEC     R5      ;TOLERANCE GAP
032256  001375                          BNE     2$      ;BRANCH IF NOT COMPLETE

                                ;THESE ARE TO SETUP FOR DISKLESS USE ONLY

032260  012737  000000  046306          MOV     #0!0,CYL        ;16 BITS PER WORD
                                                                ;CYLINDER 0, FORMAT 16 BITS
032266  112737  000001  046311          MOVB    #1,SECOTR+1     ;TRACK 1
032274  112737  000000  046310          MOVB    #0,SECOTR       ;SECTOR 0
032302  012737  000000  046312          MOV     #0,KEY1 ;KEY1=0
032310  012737  000000  046314          MOV     #0,KEY2 ;KEY2=0
032316  012737  000004  046366          MOV     #4.,DAWORD      ;NO. OF DATA WORDS
032324  005037  046316                  CLR     X               ;THIS IS A READ COMMAND
032330  004537  042666                  JSR     R5,CRC          ;GO TO CALCULATE CRC
032334  046306                          CYL
032336  050206                          WCRC

                                ;THESE ARE REGULAR SETUPS FOR RH11 AND "REINTO"

032340  004737  041460                  JSR     PC,CLDISK       ;SETUP GENERAL REGISTERS
032344  012777  177770  146660          MOV     #-4.-4.,@RHWC   ;4. DATA 4 HEADER WORDS
032352  012777  002554  146654          MOV     #REINTO,@RHBA   ;STARTING ADDRESS OF READ BUFFER
032360  112746  000000                  MOVB    #0,-(SP)        ;IN LOWER BYTE GET SECTOR
032364  112766  000001  000001          MOVB    #1,1(SP)        ;GET TRACK IN HIGHER BYTE
032372  012677  146646                  MOV     (SP)+,@RHDST    ;TRACK/SECTOR IN RHDST
032376  012777  014000  146644          MOV     #FMT22!ECI,@RHOF ;16 BITS PER WORD
                                                                ;ECC CORRECTION INHIBIT
                                                                ;BECAUSE ECC IS NOT GOING
                                                                ;TO BE CHECKED
032404  005077  146642                  CLR     @RHCA           ;CYLINDER 0

032410  004737  041514                  JSR     PC,CHECKT       ;CHECK THAT DVA,RDY,DPR,DRY = 1
032414  104401  062726                  TYPE    ,CPHALT         ;AND THAT NO OTHERS = 1.  CANNOT CON-
                                                                ;TINUE TESTING IF BOTH AREN'T TRUE
032420  000000                          HALT                    ;STOP THE TEST

032422  013711  001470                  MOV     REFOR,@R1       ;READ HEADER AND DATA=72
032426  005037  001406                  CLR     ERFLG$          ;CLEAR ERROR FLAG
032432  004737  046146                  JSR     PC,COMHD        ;READ HEADER AND DATA

                                ;IF THE PROGRAM COMES BACK HERE WITHOUT ERROR PRINTOUTS
                                ;FROM THE "COMHD" ROUTINE THAT MEANS SECTOR GAP,
                                ;FIRST SYNC, HEADER, HEADER CRC, HEADER GAP AND
                                ;SYNC BYTE HAVE GONE BY AND SYNCS WERE CORRECTLY
                                ;DETECTED.

                                ;RHWC IS CHECKED TO BE = 0 AFTER THE READ OPERATION

032436  017737  146570  001126          MOV     @RHWC,$BDDAT    ;LOAD AND TEST RHWC
032444  001401                          BEQ     20$             ;SHOULD = 0
032446  104040                          EMT     40
                                                                ;HEADER AND DATA

                                ;HEADER AND DATA ARE TO BE CHECKED.
```

```
                                        ;IN CHECKING READ DATA THE WRITE FROM BUFFER
                                        ;'WRFROM' IS FILLED WITH EXPECTED DATA AND
                                        ;COMPARISONS ARE MADE.

032450  005737  001406        20$:    TST     ERFLG$          ;ANY ERRORS ALREADY THERE
032454  001055                        BNE     TST76           ;BRANCH IF YES
032456  004737  041704                JSR     PC,CHECKE       ;CHECK THAT BITS = 1
032462  104401  062726                TYPE    ,CPHALT         ;CANNOT CONTINUE TESTING IF THEY DON'T
032466  000000                        HALT                    ;STOP THE TEST
032470  012700  001510                MOV     #WRFROM,R0      ;GETTING READY TO FILL EXPECTED DATA
032474  012720  000000                MOV     #0,(R0)+        ;CYLINDER 0
032500  112746  000000                MOVB    #0,-(SP)        ;IN LOWER BYTE GET SECTOR
032504  112766  000001  000001        MOVB    #1,1(SP)        ;GET TRACK IN HIGHER BYTE
032512  012620                        MOV     (SP)+,(R0)+     ;GET TRACK/SECTOR IN BUFFER
032514  012720  000000                MOV     #0,(R0)+        ;KEY1 IN BUFFER
032520  012720  000000                MOV     #0,(R0)+        ;KEY2 IN BUFFER
032524  012701  000400                MOV     #256.,R1        ;DATA WORD COUNTER
032530  012702  125252                MOV     #125252,R2              ;DATA
032534  010220                3$:     MOV     R2,(R0)+        ;DATA INTO BUFFER
032536  005301                        DEC     R1              ;COUNT
032540  001375                        BNE     3$              ;BRANCH IF 256 NOT DONE

                                        ;NOW READ DATA BUFFER WILL BE CHECKED

032542  004037  042354                JSR     R0,COMPAR       ;CHECK
032546  001510                        WRFROM                  ;GOOD BUFFER
032550  002554                        REINTO                  ;TEST BUFFER
032552  000010                        4+4.                    ;NUMBER OF WORDS CHECKED
032554  032562                        4$                      ;RETURN POINT FOR ERROR HEADER
032556  032566                        5$                      ;RETURN POINT FOR ERROR DATA
032560  032572                        6$                      ;RETURN FOR GOOD COMPARISON
032562                          4$:
032562  104004                        EMT     4
032564  000207                        RTS     PC              ;RETURN TO ''COMPAR''
032566                          5$:
032566  104005                        EMT     5
                                                              ;HEADER WORDS
                                                              ;5 TO 260 ARE DATA WORDS
032570  000207                        RTS     PC              ;RETURN TO ''COMPAR''

478
479                                     ;NOW SEE THAT FORMAT ERROR BIT GOT SET
480
481 032572  004737  041064        6$:     JSR     PC,PUTREG       ;SAVE REGISTERS
482
483 032576  022737  100020  001316        CMP     #FER!DCK,ER1    ;FORMAT ERROR SHOULD BE SET
484 032604  001401                        BEQ     TST76           ;BRANCH IF GOOD
485 032606  104020                        EMT     20
486                                                             ;WHEN THE DISK HAD
487                                                             ;THE FORMAT BIT=0= 18 BITS PER
488                                                             ;WORD THE READ WAS
489                                                             ;COMPLETED BUT ERROR REG
490                                                             ;WAS NOT RIGHT
491                                                             ;NOTE DCK WILL BE SET BECAUSE
492                                                             ;ECC HAS NOT BEEN GENERATED
493
500                                     ;;**************************************************************
```

```
                                        ;*TEST 76       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.
                                        ;;*****************************************************************
    032610  000004                      TST76:  SCOPE
501
502                                     ;NOW A WRITE DATA WILL BE ATTEMPTED WITH
503                                     ;WRONG FORMAT BIT
504
505 032612  012706  001100                      MOV     #STACK,SP       ;RESET STACK
506 032616  012737  000076  001432              MOV     #76,TSTNM       ;MOVE #76 TO TEST NUMBER
507
508 032624  012737  177777  046422              MOV     #-1,NOSYNC      ;SET FLAG SO THAT DATA SYNC
509                                                                     ;AND DATA IS NOT READ
510 032632  004037  041376          FRMAT1: JSR     R0,CLAREA       ;CLEAR SIMULATED DISK
511 032636  050224                          .WORD   DISK            ;FROM
512 032640  051250                          .WORD   TOLGAP+16       ;TO
513 032642  000000                          .WORD   0               ;DATA
514
515                                     ;THESE ARE SETUP FOR DISKLESS USE ONLY
516
517 032644  005037  046306                      CLR     CYL             ;CYLINDER 0, FORMAT 18 BIT WORDS
518 032650  105037  046311                      CLRB    SECOTR+1        ;TRACK 0
519 032654  105037  046310                      CLRB    SECOTR  ;SECTOR 0
520 032660  005037  046312                      CLR     KEY1            ;KEY1 0
521 032664  005037  046314                      CLR     KEY2            ;KEY2 0
522 032670  012737  000004  046354              MOV     #4,NOWORD       ;NO OF DATA WORDS
523 032676  012737  000001  046316              MOV     #1,X            ;WRITE DATA
524 032704  004537  042666                      JSR     R5,CRC  ;GO TO CALCULATE CRC
525 032710  051444                              WCYL
526 032712  051454                              GCRC
527
528                                     ;THESE AER REGULAR SETUPS
529
530 032714  004037  041376                      JSR     R0,CLAREA       ;FILL WRITE FROM BUFFER WITH 125252
531 032720  001510                              WRFROM                  ;FROM
532 032722  001516                              WRFROM+6                ;TO
533 032724  125252                              125252                  ;DATA
534 032726  004737  041460                      JSR     PC,CLDISK       ;SETUP GENERAL REGISTERS
535 032732  012777  177774  146272              MOV     #-4,@RHWC       ;256 DATA WORDS
536 032740  012777  001510  146266              MOV     #WRFROM,@RHBA   ;STARTING ADDRESS OF WRITE BUFFER
537 032746  005077  146272                      CLR     @RHDST          ;TRACK=0 SECTOR=0
538 032752  012777  010000  146270              MOV     #FMT22,@RHOF    ;16 BITS PER WORD FORMAT
539 032760  005077  146266                      CLR     @RHCA           ;CYLINDER 0
540 032764  004737  041514                      JSR     PC,CHECKT       ;CHECK THAT DVA,RDY,DPR,DRY = 1
    032770  104401  062726                      TYPE    ,CPHALT         ;AND THAT NO OTHERS = 1.  CANNOT CON-
                                                                        ;TINUE TESTING IF BOTH AREN'T TRUE
    032774  000000                              HALT                    ;STOP THE TEST
541 032776  013711  001462                      MOV     WRIDAT,@R1      ;WRITE DATA=60
542 033002  005037  001406                      CLR     ERFLG$  ;CLEAR ERROR FLAG
543 033006  004737  046146                      JSR     PC,COMHD        ;WRITE DATA
544
545                                     ;IF THE PROGRAM COMES BACK HERE WITHOUT ERROR PRINTOUTS
546                                     ;FROM THE "COMHD" ROUTINE IT MEANS SECTOR GAP, SYNC BYTE
547                                     ;HEADER, HEADER CRC, HEADER GAP AND SYNC BYTE HAVE GONE BY
```

```
                                        ;AND SYNCS WERE CORRECTLY DETECTED
                                        ;DATA IS TO BE CHECKED
033012  004737  041064                  JSR     PC,PUTREG*      ;SAVE REGISTERS
033016  005737  001406                  TST     ERFLG$  ;HAS ANY ERRORS OCCURED?
033022  001041                          BNE     4$              ;BRANCH IF YES
033024  012700  000000                  MOV     #0,R0           ;GOOD DATA
033030  012701  050224                  MOV     #DISK,R1        ;DATA WRITTEN INTO 'DISK"
033034  012702  000004                  MOV     #4,R2           ;COUNTER
033040  012737  000005  046426  1$:     MOV     #5,ERWORD       ;FOR ERROR WORD
033046  020021                          CMP     R0,(R1)+        ;COMPARE GOOD DATA WITH DATA ON DISK
033050  001424                          BEQ     3$              ;BRANCH IF GOOD
033052  010037  001124                  MOV     R0,$GDDAT       ;GOOD DATA
033056  014137  001126                  MOV     -(R1),$BDDAT    ;BAD DATA
033062  160237  046426                  SUB     R2,ERWORD       ;ERROR WORD NO
033066  005737  001406                  TST     ERFLG$  ;ANY ERRORS ALREADY THERE?
033072  001002                          BNE     2$              ;BRANCH IF YES
033074  104004                          EMT     4
                                                                ;ON A WRITE DATA WITH
                                                                ;WRONG FORMAT NO DATA
                                                                ;SHOULD BE WRITTEN
                                                                ;WORD NO GIVES WORD IN ERROR
033076  000401                          BR      5$              ;BRANCH TO AVOID PRINTING NEXT ERROR
033100                          2$:
033100  104005                          EMT     5
033102  005721                  5$:     TST     (R1)+           ;UNDO -(R1) FOR BAD DATA
033104  017746  146030                  MOV     @SWR,-(SP)      ;GET SWITCH SETTING
033110  042716  177177                  BIC     #177177,(SP)    ;KEEP ONLY SWITCH 7 AND 8
033114  022726  000200                  CMP     #SW07,(SP)+     ;IS 7 SET AND 8 RESET.
033120  001402                          BEQ     4$              ;BRANCH IF YES
033122  005302                  3$:     DEC     R2              ;IF NOT COUNT 256 WORDS
033124  001345                          BNE     1$              ;BRANCH IF 256 NOT DONE

                                        ;NOW CHECK TO SEE THAT FORMAT ERROR BIT GOT SET

033126  022737  000020  001316  4$:     CMP     #FER,ER1        ;FORMAT ERROR SHOULD BE SET
033134  001401                          BEQ     TST77           ;BRANCH IF GOOD
033136  104020                          EMT     20
                                                                ;WAS ATTEMPTED WHEN THE DISK
                                                                ;HAD THE FORMAT BIT =0=18
                                                                ;BITS PER WORD THE WRITE
                                                                ;WAS CORRECTLY ABORTED
                                                                ;BUT ERROR REG. 1 WAS WRONG

                                ;;*****************************************************************
                                ;*TEST 77        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
```

```
                                        ;;*****************************************************************
033140  000004                        TST77:  SCOPE
605 033142  012706  001100                  MOV     #STACK,SP       ;RESET STACK
606 033146  012737  000077  001432          MOV     #77,TSTNM       ;MOVE #77 TO TEST NUMBER
607 033154  004737  041460                  JSR     PC,CLDISK       ;CLEAR DISK
608 033160  004737  041514                  JSR     PC,CHECKT       ;CHECK THAT DVA,RDY,DPR,DRY = 1
033164  104401  062726                      TYPE    ,CPHALT         ;AND THAT NO OTHERS = 1.  CANNOT CON-
                                                                    ;TINUE TESTING IF BOTH AREN'T TRUE
033170  000000                              HALT                    ;STOP THE TEST
609 033172  012700  001252                  MOV     #RHCA,R0
610 033176  012005                  1$:     MOV     (R0)+,R5        ;R5 HAS ADDRESS OF REG. UNDER TEST
611 033200  052777  000040  146030  2$:     BIS     #CLR,@RHCS2
612 033206  013777  001374  146022          MOV     UNIT,@RHCS2     ;REINSTATE UNIT NO.
613
614                                     ;SET UP FOR AN OPERATION (WRITE HEADER AND DATA)
615
616 033214  013777  001464  146016          MOV     WRIFOR,@RHCS1   ;WRITE HEADER AND DATA=62
617                                                                 ;IN RHCS1
618 033222  012777  177766  146002          MOV     #-10.,@RHWC     ;10 WORDS
619 033230  012777  001510  145776          MOV     #WRFROM,@RHBA   ;BUS ADDRESS = WRFROM
620 033236  012777  000010  146000          MOV     #10,@RHDST      ;DESIRED TRACK=0, SECTOR=10
621 033244  052777  000010  145764          BIS     #BAI,@RHCS2     ;BUS ADDRESS INCREMENT INHIBIT
622 033252  012777  010000  145770          MOV     #FMT22,@RHOF    ;FORMAT 16 BIT WORDS
623 033260  005077  145766                  CLR     @RHCA           ;CYLINDER =0
624
625                                     ;SAVE REGISTERS
626
627 033264  004037  042152                  JSR     R0,SAVER        ;SAVE
628 033270  001240                          RHCS1                   ;FROM
629 033272  002554                          REINTO                  ;TO
630 033274  000016                          14.                     ;NUMBER OF REGISTERS SAVED
631
632                                     ;NOW THE COMMAND IS GIVES TO
633                                     ;WRITE HEADER AND DATA FOR CYL=0, SECTOR=10
634                                     ;TRACK=0 IT COMES BACK AFTER ONE SECTOR
635                                     ;HAS PASSED
636
637 033276  012777  000001  145754          MOV     #DMD,@RHMR      ;SET DIAGNOSTIC MODE
638 033304  005277  145730                  INC     @RHCS1          ;GO TO RHCS1 WITH 62
639 033310  012715  177672                  MOV     #177672,@R5     ;TRY WRITING ALL BITS EXCEPT
640                                                                 ;GO, RMR, IE
641 033314  052737  000001  002574          BIS     #DMD,REINTO+20  ;SET DMD IN SAVED REGISTER RHMR
642 033322  052737  000004  002556          BIS     #RMR,REINTO+2   ;SET RMR IN SAVED REG. RHER1
643 033330  042737  000200  002576          BIC     #DRY,REINTO+22  ;CLEAR DRY IN RHDS1
644 033336  052737  040000  002576          BIS     #ERR,REINTO+22  ;SET ERR IN RHDS1
645 033344  052737  000001  002554          BIS     #GO,REINTO      ;SET GO IN SAVED REG. RHCS1
646 033352  042737  000200  002554          BIC     #RDY,REINTO     ;CLEAR RDY BIT
647
648                                     ;AFTER AN ATTEMPT TO WRITE INTO A REGISTER
649                                     ;SAVE REGISTERS AGAIN SO THAT COMPARES CAN BE DONE
650
651 033360  004037  042152                  JSR     R0,SAVER        ;SAVE
652 033364  001240                          RHCS1                   ;FROM
653 033366  001510                          WRFROM                  ;TO
654 033370  000016                          14.                     ;NUMBER
655
656                                     ;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
033372  113737  002573  001527                MOVB    REINTO+17,WRFROM+17;SAVE UPPER RHAS


                                        ;COMPARE REGISTERS BEFORE ATTEMPTED WRITE WITH AFTER

033400  004037  042354                        JSR     R0,COMPAR       ;COMPAR
033404  002554                                REINTO                  ;GO BUFFER
033406  001510                                WRFROM                  ;TEST BUFFER
033410  000016                                14.                     ;NUMBER
033412  033420                                4$                      ;RETURN FOR ERROR
033414  033420                                4$                      ;SAME
033416  033440                                5$                      ;RETURN FOR GOOD COMPARISON
033420  013705  046426          4$:           MOV     ERWORD,R5       ;GETTING READY TO INDEX
033424  060505                                ADD     R5,R5           ;DOUBLE ERROR WORD
033426  016537  001236  041130                MOV     RHCS1-2(R5),REGADR ;FAILING REG. ADDRESS
033434  104001                                EMT     1
033436  000207                                RTS     PC              ;CHANGED WITH
                                                                      ;AN ATTEMPT TO WRITE
                                                                      ;DURING AN OPERATION
                                        ;THE FOLLOWING CLEAR MAY SET THE ATA BIT BECAUSE GO IS HIGH
                                        ;
033440  004737  041460          5$:           JSR     PC,CLDISK       ;CLEAR DISK


                                        ;;*****************************************************************
                                        ;*TEST 100      MAKE CURRENT CYLINDER = 1
                                        ;;*****************************************************************
033444  000004                          TST100: SCOPE
033446  012706  001100                        MOV     #STACK,SP       ;RESET STACK
033452  012737  000100  001432                MOV     #100,TSTNM      ;MOVE #100 TO TEST NUMBER
033460  004737  041460                        JSR     PC,CLDISK       ;INIT DRIVE
033464  012777  000001  145566                MOV     #DMD,@RHMR      ;SET DIAGNOSTIC MODE
033472  004037  044032                        JSR     R0,MAKECYL      ;DO SEEK COMMAND FOLLOWED BY AN INIT TO
033476  000001                                .WORD   1               ;CHANGE RHCC TO CYLINDER 1

                                        ;;*****************************************************************
                                        ;*TEST 101      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 10000,401,1,1,1, AND ALL 177400
                                        ;*AFTER THE READ THE REINTO BUFFER IS EXPECTED TO
                                        ;*HAVE WHAT IS IN WRFROM - 10000,401,1,1 AND ALL 177400
                                        ;;*****************************************************************
033500  000004                          TST101: SCOPE
033502  012706  001100                        MOV     #STACK,SP       ;RESET STACK
033506  012737  000101  001432                MOV     #101,TSTNM      ;MOVE #101 TO TEST NUMBER
033514  005037  046422                        CLR     NOSYNC          ;SET FLAG SO THAT DATA SYNC
                                                                      ;AND DATA IS  READ

                                        ;FILL SIMULATED DISK

033520  004737  043142                        JSR     PC,SETDSK       ;SET UP SIMULATED DISK
```

```
                                ;FILL REINTO BUFFER WITH 0

033524  004037  041376               JSR     R0,CLAREA       ;FILL REINTO BUFFER
033530  002554                       REINTO                  ;FROM LOCATION
033532  003554                       REINTO+<256.*2>         ;TO LOCATION
033534  000000                       0                       ;DATA

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

033536  012700  001510               MOV     #WRFROM,R0
033542  012720  010000               MOV     #FMT22,(R0)+    ;10000 INTO WRFROM
033546  012720  000401               MOV     #401,(R0)+      ;401=TRACK1,SECTOR1
033552  012720  000001               MOV     #1,(R0)+        ;1 INTO WRFROM+
033556  012720  000001               MOV     #1,(R0)+        ;1 INTO WRFROM+6

                                ;FILL ALL 0

033562  004037  041376               JSR     R0,CLAREA       ;FILL WRFROM
033566  001520                       WRFROM+10               ;FROM
033570  002510                       WRFROM+<256.*2>         ;TO
033572  177400                       177400                  ;DATA

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

033574  004037  043270               JSR     R0,HCCRCE
033600  000072                       72                      ;READ HEADER AND DATA
033602  000001                       1                       ;CYLINDER
033604  000001                       1                       ;SECTOR
033606  000001                       1                       ;TRACK
033610  177400                       -256.                   ;WORD COUNT
033612  002554                       REINTO                  ;RHBA BUFFER
033614  000000                       0                       ;READ

033616  000001                       1                       ;HEADER COMPARE
033620  000240                1$:    NOP                     ;RETURN POINT FROM HCCRCE

697
698                             ;;*****************************************************************
                                ;*TEST 102       MAKE CURRENT CYLINDER = 0
                                ;;*****************************************************************
033622  000004                TST102: SCOPE
033624  012706  001100               MOV     #STACK,SP       ;RESET STACK
033630  012737  000102  001432       MOV     #102,TSTNM      ;MOVE #102 TO TEST NUMBER
033636  004737  041460               JSR     PC,CLDISK       ;INIT DRIVE
033642  012777  000001  145410       MOV     #DMD,@RHMR      ;SET DIAGNOSTIC MODE
033650  004037  044032               JSR     R0,MAKECYL      ;DO SEEK COMMAND FOLLOWED BY AN INIT TO
033654  000000                       .WORD   0               ;CHANGE RHCC TO CYLINDER 0
699
700                             ;;*****************************************************************
                                ;*TEST 103       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
```

T103 ERROR REG1 - BIT #7 - HEADER COMPARE ERROR

```
                                ;*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
                                ;;*****************************************************************
033656  000004                  TST103: SCOPE
701 033660  012706  001100              MOV     #STACK,SP       ;RESET STACK
    033664  012737  000103  001432      MOV     #103,TSTNM      ;MOVE #103 TO TEST NUMBER
    033672  005037  046422              CLR     NOSYNC          ;SET FLAG SO THAT DATA SYNC
                                                                ;AND DATA IS  READ

                                ;FILL SIMULATED DISK

    033676  004737  043142              JSR     PC,SETDSK       ;SET UP SIMULATED DISK

                                ;FILL REINTO BUFFER WITH 0

    033702  004037  041376              JSR     R0,CLAREA       ;FILL REINTO BUFFER
    033706  002554                      REINTO                  ;FROM LOCATION
    033710  003554                      REINTO+<256.*2>         ;TO LOCATION
    033712  000000                      0                       ;DATA

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

    033714  012700  001510              MOV     #WRFROM,R0
    033720  012720  010000              MOV     #FMT22,(R0)+    ;10000 INTO WRFROM
    033724  012720  000401              MOV     #401,(R0)+      ;401=TRACK1,SECTOR1
    033730  012720  000001              MOV     #1,(R0)+        ;1 INTO WRFROM+
    033734  012720  000001              MOV     #1,(R0)+        ;1 INTO WRFROM+6

                                ;FILL ALL 0

    033740  004037  041376              JSR     R0,CLAREA       ;FILL WRFROM
    033744  001520                      WRFROM+10               ;FROM
    033746  002510                      WRFROM+<256.*2>         ;TO
    033750  177400                      177400                  ;DATA

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

    033752  004037  043270              JSR     R0,HCCRCE
    033756  000072                      72                      ;READ HEADER AND DATA
    033760  000000                      0                       ;CYLINDER
    033762  000001                      1                       ;SECTOR
    033764  000000                      0                       ;TRACK
    033766  177400                      -256.                   ;WORD COUNT
    033770  002554                      REINTO                  ;RHBA BUFFER
    033772  000000                      0                       ;READ

    033774  000001                      1                       ;HEADER COMPARE
    033776  000240              1$:     NOP                     ;RETURN POINT FROM HCCRCE
```

```
                                ;;*****************************************************************
                                ;*TEST 104       MAKE CURRENT CYLINDER = 1
                                ;;*****************************************************************
                                TST104: SCOPE
034000  000004                  
034002  012706  001100          MOV     #STACK,SP       ;RESET STACK
034006  012737  000104  001432  MOV     #104,TSTNM      ;MOVE #104 TO TEST NUMBER
034014  004737  041460          JSR     PC,CLDISK       ;INIT DRIVE
034020  012777  000001  145232  MOV     #DMD,@RHMR      ;SET DIAGNOSTIC MODE
034026  004037  044032          JSR     R0,MAKECYL      ;DO SEEK COMMAND FOLLOWED BY AN INIT TO
034032  000001                  .WORD   1               ;CHANGE RHCC TO CYLINDER 1
704
715
                                ;;*****************************************************************
                                ;*TEST 105       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=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
                                ;;*****************************************************************
034034  000004                  TST105: SCOPE
716 034036  012706  001100      MOV     #STACK,SP       ;RESET STACK
034042  012737  000105  001432  MOV     #105,TSTNM      ;MOVE #105 TO TEST NUMBER

034050  012737  177777  046422  MOV     #-1,NOSYNC      ;SET FLAG SO THAT DATA SYNC
                                                        ;AND DATA IS NOT READ

                                ;FILL SIMULATED DISK

034056  004737  043142          JSR     PC,SETDSK       ;SETUP SIMULATED DISK

                                ;FILL WRFROM WITH 125252

034062  004037  041376          JSR     R0,CLAREA       ;FILL WRFROM BUFFER
034066  001510                  WRFROM                  ;FROM LOCATION
034070  002510                  WRFROM+<256.*2>         ;TO LOCATION
034072  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

034074  004037  041376          JSR     R0,CLAREA       ;FILL REINTO BUFFER
034100  002554                  REINTO                  ;FROM LOCATION
034102  003554                  REINTO+<256.*2>         ;TO
034104  177400                  177400

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

034106  004037  043270          JSR     R0,HCCRCE
034112  000060                  60                      ;WRITE DATA
034114  000001                  1                       ;CYLINDER
```

T105 ERROR REG.1 - BIT #7 - HEADER COMPARE ERROR

```
034116  000001                          1                       ;SECTOR
034120  000001                          1                       ;TRACK
034122  177400                          -256.                   ;WORD COUNT
034124  001510                          WRFROM                  ;RHBA BUFFER
034126  000001                          1                       ;WRITE

034130  000001                          1                       ;HEADER COMPARE
034132  000240                  1$:     NOP                     ;RETURN POINT FROM HCCRCE

717
718                                     ;;*************************************************************************
                                        ;*TEST 106      MAKE CURRENT CYLINDER = 0
                                        ;;*************************************************************************
034134  000004                        TST106: SCOPE
034136  012706  001100                        MOV     #STACK,SP       ;RESET STACK
034142  012737  000106  001432                MOV     #106,TSTNM      ;MOVE #106 TO TEST NUMBER
034150  004737  041460                        JSR     PC,CLDISK       ;INIT DRIVE
034154  012777  000001  145076                MOV     #DMD,@RHMR      ;SET DIAGNOSTIC MODE
034162  004037  044032                        JSR     R0,MAKECYL      ;DO SEEK COMMAND FOLLOWED BY AN INIT TO
034166  000000                                .WORD   0               ;CHANGE RHCC TO CYLINDER 0
719
720                                     ;;*************************************************************************
                                        ;*TEST 107      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
                                        ;;*************************************************************************
034170  000004                        TST107: SCOPE
721 034172  012706  001100                      MOV     #STACK,SP       ;RESET STACK
034176  012737  000107  001432                MOV     #107,TSTNM      ;MOVE #107 TO TEST NUMBER

034204  012737  177777  046422                MOV     #-1,NOSYNC      ;SET FLAG SO THAT DATA SYNC
                                                                        ;AND DATA IS NOT READ

                                        ;FILL SIMULATED DISK

034212  004737  043142                        JSR     PC,SETDSK       ;SETUP SIMULATED DISK

                                        ;FILL WRFROM WITH 125252

034216  004037  041376                        JSR     R0,CLAREA       ;FILL WRFROM BUFFER
034222  001510                                WRFROM                  ;FROM LOCATION
034224  002510                                WRFROM+<256.*2>         ;TO LOCATION
034226  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

034230  004037  041376                        JSR     R0,CLAREA       ;FILL REINTO BUFFER
034234  002554                                REINTO                  ;FROM LOCATION
034236  003554                                REINTO+<256.*2>         ;TO
```

```
034240  177400                          177400

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

034242  004037  043270                  JSR     R0,HCCRCE
034246  000060                          60                      ;WRITE DATA
034250  000000                          0                       ;CYLINDER
034252  000001                          1                       ;SECTOR
034254  000000                          0                       ;TRACK
034256  177400                          -256.                   ;WORD COUNT
034260  001510                          WRFROM                  ;RHBA BUFFER
034262  000001                          1                       ;WRITE

034264  000001                          1                       ;HEADER COMPARE
034266  000240                  1$:     NOP                     ;RETURN POINT FROM HCCRCE

722
723
                                ;;**************************************************************
                                ;*TEST 110       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
                                ;;**************************************************************
034270  000004
724 034272  012706  001100          TST110: SCOPE
034276  012737  000110  001432              MOV     #STACK,SP       ;RESET STACK
034304  005037  046422                      MOV     #110,TSTNM      ;MOVE #110 TO TEST NUMBER
                                            CLR     NOSYNC          ;SET FLAG SO THAT DATA SYNC
                                                                    ;AND DATA IS  READ

                                ;FILL SIMULATED DISK

034310  004737  043142                      JSR     PC,SETDSK       ;SET UP SIMULATED DISK
034314  005137  050206                      COM     WCRC            ;CHANCE CRC TO GIVE HCRC

                                ;FILL REINTO BUFFER WITH 0

034320  004037  041376                      JSR     R0,CLAREA       ;FILL REINTO BUFFER
034324  002554                              REINTO                  ;FROM LOCATION
034326  003554                              REINTO+<256.*2>         ;TO LOCATION
034330  000000                              0                       ;DATA

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

034332  012700  001510                      MOV     #WRFROM,R0
034336  012720  010000                      MOV     #FMT22,(R0)+    ;10000 INTO WRFROM
034342  012720  000401                      MOV     #401,(R0)+      ;401=TRACK1,SECTOR1
034346  012720  000001                      MOV     #1,(R0)+        ;1 INTO WRFROM+
034352  012720  000001                      MOV     #1,(R0)+        ;1 INTO WRFROM+6
```

```
                                ;FILL ALL 0

034356  004037  041376                JSR     R0,CLAREA       ;FILL WRFROM
034362  001520                        WRFROM+10               ;FROM
034364  002510                        WRFROM+<256.*2>         ;TO
034366  177400                        177400                  ;DATA

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

034370  004037  043270                JSR     R0,HCCRCE
034374  000072                        72                      ;READ HEADER AND DATA
034376  000000                        0                       ;CYLINDER
034400  000001                        1                       ;SECTOR
034402  000001                        1                       ;TRACK
034404  177400                        -256.                   ;WORD COUNT
034406  002554                        REINTO                  ;RHBA BUFFER
034410  000000                        0                       ;READ

034412  000000                        0                       ;CRC ERROR
034414  000240                1$:     NOP                     ;RETURN POINT FROM HCCRCE

725
726                             ;;*****************************************************************
                                ;*TEST 111       RHER1 - BIT 8 - CRC ERROR (WRITING)
                                ;*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
                                ;;*****************************************************************
034416  000004                TST111: SCOPE
727 034420  012706  001100                MOV     #STACK,SP       ;RESET STACK
    034424  012737  000111  001432        MOV     #111,TSTNM      ;MOVE #111 TO TEST NUMBER

    034432  012737  177777  046422        MOV     #-1,NOSYNC      ;SET FLAG SO THAT DATA SYNC
                                                                    ;AND DATA IS NOT READ

                                ;FILL SIMULATED DISK

034440  004737  043142                JSR     PC,SETDSK       ;SETUP SIMULATED DISK
034444  005137  050206                COM     WCRC            ;CHANGE CRC TO GIVE HCRC

                                ;FILL WRFROM WITH 125252

034450  004037  041376                JSR     R0,CLAREA       ;FILL WRFROM BUFFER
034454  001510                        WRFROM                  ;FROM LOCATION
034456  002510                        WRFROM+<256.*2>         ;TO LOCATION
034460  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

034462  004037  041376                  JSR     R0,CLAREA       ;FILL REINTO BUFFER
034466  002554                          REINTO                  ;FROM LOCATION
034470  003554                          REINTO+<256.*2>         ;TO
034472  177400                          177400

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

034474  004037  043270                  JSR     R0,HCCRCE
034500  000060                          60                      ;WRITE DATA
034502  000000                          0                       ;CYLINDER
034504  000001                          1                       ;SECTOR
034506  000001                          1                       ;TRACK
034510  177400                          -256.                   ;WORD COUNT
034512  001510                          WRFROM                  ;RHBA BUFFER
034514  000001                          1                       ;WRITE

034516  000000                          0                       ;CRC ERROR
034520  000240                  1$:     NOP                     ;RETURN POINT FROM HCCRCE

                                ;SET UP FOR THE TWO LAST SECTOR TRANSFERRED TESTS FOLLOWING

034522  005737  001434                  TST     RP06    ;MOVE RP06 FLAG TO ITSELF TO TEST
034526  001401                          BEQ     2$              ;IF = 0 TREAT DRIVE AS RP04
034530  000402                          BR      3$              ;TREAT AS RP06 - DO NEXT 'MAKECL" & TEST
034532  000137  035206          2$:     JMP     DOG             ;DO SECOND FOLLOWING 'MAKECL" AND TEST
034536                          3$:

                                ;;*****************************************************************
                                ;*TEST 112       MAKE CURRENT CYLINDER = 814.
                                ;;*****************************************************************
034536  000004                  TST112: SCOPE
034540  012706  001100                  MOV     #STACK,SP       ;RESET STACK
034544  012737  000112  001432          MOV     #112,TSTNM      ;MOVE #112 TO TEST NUMBER
034552  004737  041460                  JSR     PC,CLDISK       ;INIT DRIVE
034556  012777  000001  144474          MOV     #DMD,@RHMR      ;SET DIAGNOSTIC MODE
034564  004037  044032                  JSR     R0,MAKECYL      ;DO SEEK COMMAND FOLLOWED BY AN INIT TO
034570  001456                          .WORD   814.            ;CHANGE RHCC TO CYLINDER 814.

                                ;;*****************************************************************
                                ;*TEST 113       RHDS1 (BIT #10) - LAST SECTOR TRANSFERRED, 'LST'
                                ;*WRITE CYLINDER 814., 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.
                                ;;*****************************************************************
034572  000004                  TST113: SCOPE
034574  012706  001100                  MOV     #STACK,SP       ;RESET STACK
034600  012737  000113  001432          MOV     #113,TSTNM      ;MOVE #113 TO TEST NUMBER
```

```
034606  004037  041376                  JSR     R0,CLAREA       ;CLEAR SIMULATED DISK
034612  050224                          .WORD   DISK            ;FROM
034614  051250                          .WORD   TOLGAP+16       ;TO
034616  000000                          .WORD   0               ;DATA

                                ;THESE ARE SETUP FOR DISKLESS USE ONLY

034620  012737  011456  046306          MOV     #814.!FMT22,CYL ;CYLINDER 814.
                                                                ;16 BITS PER WORD
034626  112737  000022  046311          MOVB    #18.,SECOTR+1   ;TRACK 18.
034634  112737  000025  046310          MOVB    #21.,SECOTR     ;SECTOR 21.
034642  005037  046312                  CLR     KEY1            ;KEY1 0
034646  005037  046314                  CLR     KEY2            ;KEY2 0
034652  012737  000400  046354          MOV     #256.,NOWORD    ;NO OF DATA WORDS
034660  012737  000001  046316          MOV     #1,X            ;WRITE DATA
034666  004537  042666                  JSR     R5,CRC          ;GO TO CALCULATE CRC
034672  046306                          CYL
034674  050206                          WCRC

                                ;THESE ARE REGULAR SETUPS

034676  004037  041376                  JSR     R0,CLAREA       ;FILL WRITE BUFFER WITH 377
034702  001510                          WRFROM                  ;FROM LOCATION
034704  002510                          WRFROM+<256.*2>         ;TO LOCATION
034706  000377                          377                     ;DATA
034710  004737  041460                  JSR     PC,CLDISK       ;SETUP GENERAL REGISTERS
034714  012777  177400  144310          MOV     #-256.,@RHWC    ;256. DATA WORDS
034722  012777  001510  144304          MOV     #WRFROM,@RHBA   ;STARTING ADDRESS OF WRITE BUFFER
034730  012746  000025                  MOV     #21.,-(SP)      ;SECTOR 21.
034734  112766  000022  000001          MOVB    #18.,1(SP)      ;TRACK 18.
034742  012677  144276                  MOV     (SP)+,@RHDST    ;SECTOR 21. TRACK 18.
034746  012777  010000  144274          MOV     #FMT22,@RHOF    ;16 BITS PER WORD FORMAT
034754  012777  001456  144270          MOV     #814.,@RHCA     ;CYLINDER 814.
034762  004737  041514                  JSR     PC,CHECKT       ;CHECK THAT DVA,RDY,DPR,DRY = 1
034766  104401  062726                  TYPE    ,CPHALT         ;AND THAT NO OTHERS = 1.  CANNOT CON-
                                                                ;TINUE TESTING IF BOTH AREN'T TRUE
034772  000000                          HALT                    ;STOP THE TEST
034774  013711  001462                  MOV     WRIDAT,@R1      ;WRITE DATA=60
035000  005037  001406                  CLR     ERFLG$          ;CLEAR ERROR FLAG
035004  004737  046146                  JSR     PC,COMHD        ;WRITE DATA

                        ;IF THE PROGRAM COMES BACK HERE WITHOUT ERROR PRINTOUTS
                        ;FROM THE "COMHD" ROUTINE IT MEANS SECTOR GAP, SYNC BYTE
                        ;HEADER, HEADER CRC, HEADER GAP AND SYNC BYTE HAVE GONE BY
                        ;AND SYNCS WERE CORRECTLY DETECTED, DATA IS TO BE CHECKED.

035010  004737  041064          JSR     PC,PUTREG       ;SAVE REGISTERS
035014  005737  001406          TST     ERFLG$          ;HAVE ANY ERRORS OCCURED?
035020  001062                  BNE     5$              ;BRANCH IF YES
035022  012700  000377          MOV     #377,R0         ;GOOD DATA
035026  012701  050224          MOV     #DISK,R1        ;DATA WRITTEN INTO "DISK"
035032  012702  000400          MOV     #256.,R2        ;COUNTER

035036  012737  000401  046426  1$:     MOV     #256.+1,ERWORD  ;FOR ERROR WORD
035044  020021                  CMP     R0,(R1)+        ;COMPARE GOOD DATA WITH DATA ON DISK
035046  001424                  BEQ     3$              ;BRANCH IF GOOD
035050  010037  001124          MOV     R0,$GDDAT       ;GOOD DATA
```

```
035054  014137  001126                  MOV     -(R1),$BDDAT    ;BAD DATA
035060  160237  046426                  SUB     R2,ERWORD       ;ERROR WORD NO
035064  005737  001406                  TST     ERFLG$          ;ANY ERRORS ALREADY THERE?
035070  001002                          BNE     2$              ;BRANCH IF YES
035072  104004                          EMT     4
035074  000401                          BR      1064$           ;BRANCH TO AVOID PRINTING NEXT ERROR

035076                          2$:
035076  104005                          EMT     5
035100  005721                  1064$:  TST     (R1)+           ;UNDO -(R1) FOR BAD DATA
035102  017746  144032                  MOV     @SWR,-(SP)      ;GET SWITCH SETTING
035106  042716  177177                  BIC     #177177,(SP)    ;KEEP ONLY SWITCH 7 AND 8
035112  022726  000200                  CMP     #SW07,(SP)+     ;IS 7 SET AND 8 RESET
035116  001402                          BEQ     4$              ;BRANCH OUT IF YES
035120  005302                  3$:     DEC     R2              ;IF NOT COUNT 256 WORDS
035122  001345                          BNE     1$              ;BRANCH IF 256. NOT DONE

748
749 035124  013746  001336          4$:     MOV     DS1,-(SP)       ;GET RHDS1
750 035130  042716  001000                  BIC     #PROG,(SP)      ;CLEAR PROG
751 035134  022726  002700                  CMP     #LST!DPR!DRY!VV,(SP)+;IS 'LST' HIGH ?
752 035140  001412                          BEQ     5$              ;BRANCH IF GOOD
753 035142  013737  001262  041130          MOV     RHDS1,REGADR ;FAILING REG. ADDRESS
754 035150  012737  002700  001124          MOV     #LST!DPR!DRY!VV,$GDDAT ;GOOD DATA
755 035156  013737  001336  001126          MOV     DS1,$BDDAT      ;BAD DATA
756 035164  104001                          EMT     1
757                                                     ;LAST SECTOR ON LAST TRACK
758                                                     ;ON LAST CYLINDER WAS
759                                                     ;WRITTEN
760                                                     ;VV BIT #6 MAY OR MAY NOT BE HIGH
761 035166  013737  001240  035176  5$:     MOV     RHCS1,6$        ;SET UP "WAT" SUBROUTINE
762 035174  104415                          WAT
763 035176  000000                  6$:     0                       ;RHCS1 ADDRESS
764 035200  000200                          RDY                     ;WAIT FOR READY
765
766 035202  000137  035652                  JMP     CAT             ;DON'T DO THE RP04 'LST' TEST FOLLOWING
767
768 035206                          DOG:
                                    ;;*****************************************************************
                                    ;*TEST 114      MAKE CURRENT CYLINDER = 410.
                                    ;;*****************************************************************
    035206  000004                  TST114: SCOPE
    035210  012706  001100                  MOV     #STACK,SP       ;RESET STACK
    035214  012737  000114  001432          MOV     #114,TSTNM      ;MOVE #114 TO TEST NUMBER
    035222  004737  041460                  JSR     PC,CLDISK       ;INIT DRIVE
    035226  012777  000001  144024          MOV     #DMD,@RHMR      ;SET DIAGNOSTIC MODE
    035234  004037  044032                  JSR     R0,MAKECYL      ;DO SEEK COMMAND FOLLOWED BY AN INIT TO
    035240  000632                          .WORD   410.            ;CHANGE RHCC TO CYLINDER 410.
769
777
                                    ;;*****************************************************************
                                    ;*TEST 115      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 THE
                                    ;*WRITE IS COMPLETED
                                    ;;*****************************************************************
```

```
035242  000004                          TST115: SCOPE
   778  035244  012706  001100                  MOV     #STACK,SP       ;RESET STACK
        035250  012737  000115  001432          MOV     #115,TSTNM      ;MOVE #115 TO TEST NUMBER

        035256  004037  041376                  JSR     R0,CLAREA       ;CLEAR SIMULATED DISK
        035262  050224                          .WORD   DISK            ;FROM
        035264  051250                          .WORD   TOLGAP+16       ;TO
        035266  000000                          .WORD   0               ;DATA

                                        ;THESE ARE SETUP FOR DISKLESS USE ONLY

        035270  012737  010632  046306          MOV     #410.!FMT22,CYL ;CYLINDER 410.
                                                                        ;16 BITS PER WORD
        035276  112737  000022  046311          MOVB    #18.,SECOTR+1   ;TRACK 18.
        035304  112737  000025  046310          MOVB    #21.,SECOTR     ;SECTOR 21.
        035312  005037  046312                  CLR     KEY1            ;KEY1 0
        035316  005037  046314                  CLR     KEY2            ;KEY2 0
        035322  012737  000400  046354          MOV     #256.,NOWORD    ;NO OF DATA WORDS
        035330  012737  000001  046316          MOV     #1,X            ;WRITE DATA
        035336  004537  042666                  JSR     R5,CRC          ;GO TO CALCULATE CRC
        035342  046306                          CYL
        035344  050206                          WCRC

                                        ;THESE ARE REGULAR SETUPS

        035346  004037  041376                  JSR     R0,CLAREA       ;FILL WRITE BUFFER WITH 377
        035352  001510                          WRFROM                  ;FROM LOCATION
        035354  002510                          WRFROM+<256.*2>         ;TO LOCATION
        035356  000377                          377                     ;DATA
        035360  004737  041460                  JSR     PC,CLDISK       ;SETUP GENERAL REGISTERS
        035364  012777  177400  143640          MOV     #-256.,@RHWC    ;256. DATA WORDS
        035372  012777  001510  143634          MOV     #WRFROM,@RHBA   ;STARTING ADDRESS OF WRITE BUFFER
        035400  012746  000025                  MOV     #21.,-(SP)      ;SECTOR 21.
        035404  112766  000022  000001          MOVB    #18.,1(SP)      ;TRACK 18.
        035412  012677  143626                  MOV     (SP)+,@RHDST    ;SECTOR 21. TRACK 18.
        035416  012777  010000  143624          MOV     #FMT22,@RHOF    ;16 BITS PER WORD FORMAT
        035424  012777  000632  143620          MOV     #410.,@RHCA     ;CYLINDER 410.
        035432  004737  041514                  JSR     PC,CHECKT       ;CHECK THAT DVA,RDY,DPR,DRY = 1
        035436  104401  062726                  TYPE    ,CPHALT         ;AND THAT NO OTHERS = 1.  CANNOT CON-
                                                                        ;TINUE TESTING IF BOTH AREN'T TRUE
        035442  000000                          HALT                    ;STOP THE TEST
        035444  013711  001462                  MOV     WRIDAT,@R1      ;WRITE DATA=60
        035450  005037  001406                  CLR     ERFLG$          ;CLEAR ERROR FLAG
        035454  004737  046146                  JSR     PC,COMHD        ;WRITE DATA

                                        ;IF THE PROGRAM COMES BACK HERE WITHOUT ERROR PRINTOUTS
                                        ;FROM THE "COMHD" ROUTINE IT MEANS SECTOR GAP, SYNC BYTE
                                        ;HEADER, HEADER CRC, HEADER GAP AND SYNC BYTE HAVE GONE BY
                                        ;AND SYNCS WERE CORRECTLY DETECTED, DATA IS TO BE CHECKED.

        035460  004737  041064                  JSR     PC,PUTREG       ;SAVE REGISTERS
        035464  005737  001406                  TST     ERFLG$          ;HAVE ANY ERRORS OCCURED?
        035470  001062                          BNE     5$              ;BRANCH IF YES
        035472  012700  000377                  MOV     #377,R0         ;GOOD DATA
        035476  012701  050224                  MOV     #DISK,R1        ;DATA WRITTEN INTO "DISK"
        035502  012702  000400                  MOV     #256.,R2        ;COUNTER
```

```
035506  012737  000401  046426  1$:     MOV     #256.+1,ERWORD  ;FOR ERROR WORD
035514  020021                          CMP     R0,(R1)+        ;COMPARE GOOD DATA WITH DATA ON DISK
035516  001424                          BEQ     3$              ;BRANCH IF GOOD
035520  010037  001124                  MOV     R0,$GDDAT       ;GOOD DATA
035524  014137  001126                  MOV     -(R1),$BDDAT    ;BAD DATA
035530  160237  046426                  SUB     R2,ERWORD       ;ERROR WORD NO
035534  005737  001406                  TST     ERFLG$          ;ANY ERRORS ALREADY THERE?
035540  001002                          BNE     2$              ;BRANCH IF YES
035542  104004                          EMT     4
035544  000401                          BR      1064$           ;BRANCH TO AVOID PRINTING NEXT ERROR

035546                          2$:
035546  104005                          EMT     5
035550  005721                  1064$:  TST     (R1)+           ;UNDO -(R1) FOR BAD DATA
035552  017746  143362                  MOV     @SWR,-(SP)      ;GET SWITCH SETTING
035556  042716  177177                  BIC     #177177,(SP)    ;KEEP ONLY SWITCH 7 AND 8
035562  022726  000200                  CMP     #SW07,(SP)+     ;IS 7 SET AND 8 RESET
035566  001402                          BEQ     4$              ;BRANCH OUT IF YES
035570  005302                  3$:     DEC     R2              ;IF NOT COUNT 256 WORDS
035572  001345                          BNE     1$              ;BRANCH IF 256. NOT DONE

779
780 035574  013746  001336          4$:     MOV     DS1,-(SP)       ;GET RHDS1
781 035600  042716  001000                  BIC     #PROG,(SP)      ;CLEAR PROG BIT
782 035604  022726  002700                  CMP     #LST!DPR!DRY!VV,(SP)+  ;IS 'LST' HIGH ?
783 035610  001412                          BEQ     5$              ;WAIT FOR 'RDY' IF GOOD
784 035612  013737  001262  041130          MOV     RHDS1,REGADR ;FAILING REG. ADDRESS
785 035620  012737  002700  001124          MOV     #LST!DPR!DRY!VV,$GDDAT ;GOOD DATA
786 035626  013737  001336  001126          MOV     DS1,$BDDAT      ;BAD DATA
787 035634  104001                          EMT     1
788                                                                 ;LAST SECTOR ON LAST TRACK ON LAST
789                                                                 ;CYLINDER WAS WRITTEN - 'VV' BIT #6
790                                                                 ;MAY OR MAY NOT BE HIGH
791
792 035636  013737  001240  035646  5$:     MOV     RHCS1,6$        ;SET UP 'WAT'' SUBROUTINE
793 035644  104415                          WAT
794 035646  000000                  6$:     0                       ;RHCS1 ADDRESS
795 035650  000200                          RDY                     ;WAIT FOR 'RDY' BIT
796
812                                 ;;*****************************************************************
                                    ;*TEST 116       ERROR REGISTER 1 - BIT #9 AOE
                                    ;*A WRITE DATA COMMAND IS GIVEN TO CYLINDER 410./814.
                                    ;*SECTOR 21 TRACK 18, KEYS 0, DATA 377
                                    ;*WORD COUNT REGISTER FOR 522 (256+66+200) WORDS
                                    ;*
                                    ;*AFTER 256 WORDS HAVE BEEN WRITTEN
                                    ;*'AOE' SHOULD COME UP

                                    ;*RHWC WILL SHOW 200 BECAUSE THE SILO IS 66 WORDS AND
                                    ;*256 WORDS HAVE BEEN WRITTEN - TOTAL 322
                                    ;*THIS IS 200 SHORT OF 522
                                    ;;*****************************************************************
    035652                          TST116:
    035652                          CAT:
813 035652  012706  001100                  MOV     #STACK,SP       ;RESET STACK
814 035656  012737  000116  001432          MOV     #116,TSTNM      ;MOVE #116 TO TEST NUMBER
815 035664  004737  041460                  JSR     PC,CLDISK       ;INIT AND SET UP GENERAL REG. CORRES.
```

```
  816 035670  004037  041376                    JSR     R0,CLAREA       ;CLEAR SIMULATED DISK
  817 035674  050224                            .WORD   DISK            ;FROM
  818 035676  051250                            .WORD   TOLGAP+16       ;TO
  819 035700  000000                            .WORD   0               ;DATA
  820
  821                                   ;THESE ARE TO SETUP FOR DISKLESS USE ONLY
  822                                   ;AND WILL HANDLE RP04 OR RP06 DRIVES
  823
  824 035702  005737  001434                    TST     RP06    ;MOVE RP06 FLAG TO ITSELF TO TEST
  825 035706  001404                            BEQ     10$             ;TREAT DRIVE AS RP04 IF = 0
  826
  827 035710  012737  011456  046306            MOV     #814.!FMT22,CYL;CYLINDER 814., 16 BITS PER WORD
  828 035716  000403                            BR      11$             ;TREAT DRIVE AS RP06
  829
  830 035720  012737  010632  046306  10$:      MOV     #410.!FMT22,CYL;CYLINDER 410., 16 BITS PER WORD
  831                                                                   ;TREAT DRIVE AS RP04
  832
  833 035726  112737  000022  046311  11$:      MOVB    #18.,SECOTR+1   ;TRACK 18.
  834 035734  112737  000025  046310            MOVB    #21.,SECOTR     ;SECTOR 21.
  835 035742  005037  046312                    CLR     KEY1            ;KEY1 0
  836 035746  005037  046314                    CLR     KEY2            ;KEY2 0
  837 035752  012737  000400  046354            MOV     #256.,NOWORD    ;NO OF DATA WORDS
  838 035760  012737  000001  046316            MOV     #1,X            ;WRITE DATA
  839 035766  004537  042666                    JSR     R5,CRC  ;GO TO CALCULATE CRC
  840 035772  046306                            CYL
  841 035774  050206                            WCRC
  842
  843                                   ;THESE ARE REGULAR SETUPS
  844
  845 035776  004037  041376                    JSR     R0,CLAREA       ;FILL WRITE BUFFER WITH 377
  846 036002  001510                            WRFROM                  ;FROM
  847 036004  002510                            WRFROM+<256.*2>         ;TO
  848 036006  000377                            377                     ;DATA
  849 036010  004737  041460                    JSR     PC,CLDISK       ;SETUP GENERAL REGISTERS
  850 036014  012777  176766  143210            MOV     #-522.,@RHWC    ;522. DATA WORDS
  851 036022  012777  001510  143204            MOV     #WRFROM,@RHBA   ;STARTING ADDRESS OF WRITE BUFFER
  852 036030  012746  000025                    MOV     #21.,-(SP)      ;SECTOR 21.
  853 036034  112766  000022  000001            MOVB    #18.,1(SP)      ;TRACK 18.
  854 036042  012677  143176                    MOV     (SP)+,@RHDST    ;SECTOR 21. TRACK 18.
  855 036046  012777  010000  143174            MOV     #FMT22,@RHOF    ;16 BITS PER WORD FORMAT
  856
  857                                   ;CHECK TO SEE WHAT TYPE OF DRIVE IS BEING TESTED
  858                                   ;AND LOAD CYLINDER ADDRESS REGISTER WITH THE PROPER NUMBER
  859
  860 036054  005737  001434                    TST     RP06    ;MOVE FLAG TO ITSELF TO TEST
  861 036060  001404                            BEQ     12$             ;TREAT AS RP04 IF = 0
  862 036062  012777  001456  143162            MOV     #814.,@RHCA     ;CYLINDER 814.
  863 036070  000403                            BR      13$             ;TREAT AS RP06
  864 036072  012777  000632  143152  12$:      MOV     #410.,@RHCA     ;CYLINDER 410.
  865 036100                          13$:
  866
  867 036100  004737  041514                    JSR     PC,CHECKT       ;CHECK THAT DVA,RDY,DPR,DRY = 1
      036104  104401  062726                    TYPE    ,CPHALT         ;AND THAT NO OTHERS = 1.  CANNOT CON-
                                                                        ;TINUE TESTING IF BOTH AREN'T TRUE
      036110  000000                            HALT                    ;STOP THE TEST
  868 036112  013711  001462                    MOV     WRIDAT,@R1      ;WRITE DATA=60
  869 036116  005037  001406                    CLR     ERFLG$  ;CLEAR ERROR FLAG
```

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

036122  004037  042152                      JSR     R0,SAVER        ;SAVE
036126  001232                              RHWC                    ;FROM
036130  002554                              REINTO                  ;TO
036132  000023                              19.                     ;NUMBER SAVED

                                        ;GIVE WRITE DATA COMMAND

036134  004737  046146                      JSR     PC,COMHD        ;WRITE DATA COMMAND

                                        ;CHANGE SAVED REGISTERS TO EXPECTED VALUES

036140  005737  001436                      TST     RH70            ;CHECK FOR RH70 CONTROLLER
036144  001407                              BEQ     JP1             ;SKIP RH70 CODE AND DO RH11 IF NOT

036146  012737  177376  002554  VAR1:       MOV     #-258.,REINTO   ;SAVED RHWC SHOULD BE = 258.
036154  012737  002530  002556  VAR2:       MOV     #WRFROM+<2*256.>+<2*8.>,REINTO+2
                                                                    ;SAVED RHBA SHOULD BE WRFROM+256+8
036162  000406                              BR      JP2             ;SKIP NEXT RH11 CODE

036164  012737  177470  002554  JP1:        MOV     #-200.,REINTO   ;SAVED RHWC SHOULD BE = 200.
036172  012737  002714  002556              MOV     #WRFROM+<2*256.>+<2*66.>,REINTO+2
                                                                    ;SAVED RHBA SHOULD BE WRFROM+256+66

036200  052737  000200  002560  JP2:        BIS     #OR,REINTO+4    ;SAVED RHCS2
036206  042737  000100  002560              BIC     #IR,REINTO+4    ;SAVED RHCS2
036214  052737  140000  002562              BIS     #SC!TRE,REINTO+6;SAVED RHCS1 SHOULD HAVE 'SC' & 'TRE'
036222  012737  001000  002564              MOV     #AOE,REINTO+10  ;SAVED RHER1 SHOULD HAVE 'AOE'
036230  017737  143010  002566              MOV     @RHDST,REINTO+12;SAVED RHDST SHOULD HAVE=
                                                                    ;RHDST IS UNDEFINED

                                        ;CHECK TO SEE WHAT TYPE OF DRIVE IS BEING TESTED
                                        ;AND SET UP CYLINDER ADDRESS ACCORDINGLY

036236  005737  001434                          TST     RP06    ;MOVE RP06 FLAG TO ITSELF TO TEST
036242  001404                                  BEQ     14$             ;TREAT AS RP04 IF = 0
036244  012737  001457  002574                  MOV     #815.,REINTO+20;SAVED DESIRED CYLINDER ADDRESS
036252  000403                                  BR      15$             ;TREAT AS RP06
036254  012737  000633  002574  14$:            MOV     #411.,REINTO+20;SAVED DESIRED CYLINDER ADDRESS

036262  013737  001416  002600  15$:        MOV     ATTENT,REINTO+24 ;SAVED RHAS SHOULD HAVE APPRO. BIT
036270  052737  000001  002602              BIS     #DMD,REINTO+26;SAVED RHMR
036276  052737  142000  002604              BIS     #ATA!ERR!LST,REINTO+30 ;SAVED RHDS1

                                        ;AFTER A WRITE DATA COMMAND WITH 'AOE' ERROR
                                        ;SAVE REGISTERS AGAIN SO THAT COMPARES CAN BE DONE

036304  004037  042152                      JSR     R0,SAVER        ;SAVE
036310  001232                              RHWC                    ;FROM
036312  001510                              WRFROM                  ;TO
036314  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
```

```
                                        MOVB    REINTO+25,WRFROM+25;SAVE UPPER RHAS

                                ;COMPARE REGISTERS BEFORE WRITE DATA COMMAND
                                ;WITH AFTER COMMAND

                                        JSR     R0,COMPAR       ;COMPARE
                                        REINTO                  ;GOOD BUFFER
                                        WRFROM                  ;TEST BUFFER
                                        17.                     ;NUMBER OF REGISTERS
                                        1$                      ;RETURN FOR ERROR
                                        1$                      ;SAME
                                        2$                      ;RETURN FOR GOOD COMPARISON

                                1$:     MOV     ERWORD,R5       ;GETTING READY TO INDEX
                                        ADD     R5,R5           ;DOUBLE ERROR WORD
                                        MOV     RHWC-2(R5),REGADR ;FAILING REG. ADDRESS
                                        EMT     1
                                                                ;REGISTER CHANGE
                                        RTS     PC              ;RETURN FOR FURTHER COMPARISONS
                                                                ;NO ERRORS
                                2$:     CLR     ERFLG$  ;CLEAR ERROR FLAG


                                ;DATA IS TO BE CHECKED HERE

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

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

                                7$:
                                ;;*************************************************************
                                ;*TEST 117      MAKE CURRENT CYLINDER = 0
                                ;;*************************************************************
                                TST117: SCOPE
                                        MOV     #STACK,SP       ;RESET STACK
```

```
T117    MAKE CURRENT CYLINDER = 0

        036504  012737  000117  001432          MOV     #117,TSTNM      ;MOVE #117 TO TEST NUMBER
        036512  004737  041460                  JSR     PC,CLDISK       ;INIT DRIVE
        036516  012777  000001  142534          MOV     #DMD,@RHMR      ;SET DIAGNOSTIC MODE
        036524  004037  044032                  JSR     R0,MAKECYL      ;DO SEEK COMMAND FOLLOWED BY AN INIT TO
        036530  000000                          .WORD   0               ;CHANGE RHCC TO CYLINDER 0
    979
    987                                 ;;*****************************************************************
                                        ;*TEST 120      ERROR REGISTER 1 - BIT #10 'IAE'
                                        ;*A READ HEADER AND DATA IS GIVEN TO TRACK 20, SECTOR 0
                                        ;*
                                        ;*AN INDEX PULSE IS GIVEN TO GET RHLA TO 0
                                        ;*
                                        ;*IAE BIT SHOULD SET
                                        ;;*****************************************************************
        036532  000004                  TST120: SCOPE
    988 036534  012706  001100                  MOV     #STACK,SP       ;RESET STACK
    989 036540  012737  000120  001432          MOV     #120,TSTNM      ;MOVE #120 TO TEST NUMBER
    990 036546  004737  041460                  JSR     PC,CLDISK       ;CLEAR REGISTERS AND SET UNIT NO.
    991
    992                                 ;GIVE INDEX PULSE
    993 036552  012777  000001  142500          MOV     #DMD,@RHMR      ;SET DIAGNOSTIC MODE
    994 036560  052777  000004  142472          BIS     #MINX,@RHMR     ;SET INDEX
    995 036566  042777  000004  142464          BIC     #MINX,@RHMR     ;CLEAR INDEX
    996
    997
    998                                 ;THESE ARE REGULAR SETUPS
    999
   1000 036574  012777  177400  142430          MOV     #-256.,@RHWC    ;256 DATA  WORDS 4 HEADER WORDS
   1001 036602  012700  002554                  MOV     #REINTO,R0      ;THESE TWO INSTRUCTIONS GETS
   1002 036606  010077  142422                  MOV     R0,@RHBA        ;ADDR, OF WRFROM INTO R0 AND
   1003                                                                 ;BUS ADDRESS REGISTER
   1004 036612  012720  010000                  MOV     #FMT22,(R0)+    ;FORMAT=16 BIT WORDS
   1005                                                                 ;CYLINDER=0
   1006 036616  012720  012000                  MOV     #12000,(R0)+    ;TRACK=20 SECTOR=0 KEYS=0
   1007 036622  005020                          CLR     (R0)+           ;KEY1=0
   1008 036624  005020                          CLR     (R0)+           ;KEY2=0
   1009 036626  012705  000400                  MOV     #256.,R5        ;COUNTER
   1010 036632  012720  177777          1$:     MOV     #-1,(R0)+       ;MOVE ALL ONES FOR DATA
   1011 036636  005305                          DEC     R5
   1012 036640  001374                          BNE     1$              ;BRANCH IF DATA NOT COMPLETE
   1013 036642  012777  012000  142374          MOV     #12000,@RHDST   ;TRACK=20 SECTOR=0
   1014
   1015 036650  004737  041514                  JSR     PC,CHECKT       ;CHECK THAT DVA,RDY,DPR,DRY = 1
        036654  104401  062726                  TYPE    ,CPHALT         ;AND THAT NO OTHERS = 1.  CANNOT CON-
                                                                        ;TINUE TESTING IF BOTH AREN'T TRUE
        036660  000000                          HALT                    ;STOP THE TEST
   1016
   1017 036662  013711  001470                  MOV     REFOR,@R1       ;GET READY FOR WRITE HEADER AND
   1018                                                                 ;DATA WITH 62 IN RHCS1
   1019 036666  005037  001406                  CLR     ERFLG$  ;CLEAR ERROR FLAG
   1020 036672  012777  010000  142350          MOV     #FMT22,@RHOF    ;FORMAT BIT=1 (16 BIT WORDS)
   1021 036700  005077  142346                  CLR     @RHCA           ;CYLINDER =0
   1022
   1023                                 ;THE REGISTERS WILL BE SAVED IN REINTO BUFFER
   1024 036704  004037  042152                  JSR     R0,SAVER        ;SAVE
   1025 036710  001232                          RHWC                    ;FROM
   1026 036712  002554                          REINTO                  ;TO
```

```
1027 036714  000023                        19.                     ;NUMBER SAVED
1028
1029                                ;GO TO WRITE HEADER AND DATA
1030
1031 036716  013700  001260                MOV     RHMR,R0 ;NOW R0 WAS MAINTENANCE REG. ADDR.
1032 036722  012710  000001                MOV     #DMD,@R0        ;SET DIAGNOSTIC MODE
1033 036726  052777  000001  142304        BIS     #GO,@RHCS1      ;GO
1034
1035                                ;CHANGE SAVED REGISTERS TO EXPECTED VALUE
1036 036734  052737  140000  002562        BIS     #SC!TRE,REINTO+6 ;SAVED RHCS1
1037 036742  012737  002000  002564        MOV     #IAE,REINTO+10 ;SAVED RHER1
1038 036750  012737  012001  002566        MOV     #12001,REINTO+12;SAVED RHDST
1039 036756  013737  001416  002600        MOV     ATTENT,REINTO+24 ;SAVED RHAS
1040 036764  052737  000001  002602        BIS     #DMD,REINTO+26 ;SAVED RHMR
1041 036772  052737  140000  002604        BIS     #ATA!ERR,REINTO+30 ;SAVED RHDS1
1042
1043                                ;SAVE REGISTERS AGAIN SO THAT COMPARES CAN BE DONE
1044 037000  004037  042152                JSR     R0,SAVER        ;SAVE
1045 037004  001232                        RHWC                    ;FROM
1046 037006  001510                        WRFROM                  ;TO
1047 037010  000023                        19.                     ;NUMBER OF REGISTERS SAVED
1048
1049                                ;AS UPPER BYTE OF RHAS CAN BE CHANGING IN A DUAL PORT
1050                                ;OPERATION THE UPPER BYTE OF RHAS WILL BE SAVED AS IS
1051                                ;SO THAT THE COMPARES ARE ONLY VALID FOR THE LOWER BYTE
1052 037012  113737  002601  001535        MOVB    REINTO+25,WRFROM+25;SAVE UPPER RHAS
1053
1054
1055                                ;COMPARE REGISTERS BEFORE READ IN PRESET COMMAND
1056                                ;WITH AFTER COMMAND
1057 037020  004037  042354                JSR     R0,COMPAR       ;COMPARE
1058 037024  002554                        REINTO                  ;GOOD BUFFER
1059 037026  001510                        WRFROM                  ;TEST BUFFER
1060 037030  000021                        17.                     ;NUMBER OF REGISTERS
1061 037032  037040                        2$                      ;RETURN FOR ERROR
1062 037034  037040                        2$                      ;SAME
1063 037036  037060                        3$                      ;RETURN FOR GOOD COMPARISON
1064
1065 037040  013705  046426        2$:     MOV     ERWORD,R5       ;GETTING READY TO INDEX
1066 037044  060505                        ADD     R5,R5           ;DOUBLE ERROR WORD
1067 037046  016537  001230  041130        MOV     RHWC-2(R5),REGADR ;FAILING REG. ADDRESS
1068 037054  104001                        EMT     1
1069                                                                ;REGISTER CHANGE
1070 037056  000207                        RTS     PC              ;RETURN FOR FURTHER COMPARISONS
1071
1072                                                                ;NO ERRORS
1073
1074 037060  004737  041460        3$:     JSR     PC,CLDISK       ;CLEAR GO BIT
1075
1086                                ;;*****************************************************************
                                    ;*TEST 121      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
                                        ;*********************************************************************
     037064  000004                     TST121: SCOPE
1087 037066  012706  001100                     MOV     #STACK,SP       ;RESET STACK
1088 037072  012737  000121  001432            MOV     #121,TSTNM      ;MOVE #121 TO TEST NUMBER
1089 037100  004737  041460                     JSR     PC,CLDISK       ;CLEAR REGISTERS AND SET UNIT NO.
1090
1091                                    ;GIVE INDEX PULSE
1092 037104  012777  000001  142146            MOV     #DMD,@RHMR      ;SET DIAGNOSTIC MODE
1093 037112  052777  000004  142140            BIS     #MINX,@RHMR     ;SET INDEX
1094 037120  042777  000004  142132            BIC     #MINX,@RHMR     ;CLEAR INDEX
1095
1096                                    ;THESE ARE REGULAR SETUPS
1097
1098 037126  012777  177400  142076            MOV     #-256.,@RHWC    ;256 DATA WORDS 4 HEADER WORDS
1099 037134  012700  001510                     MOV     #WRFROM,R0      ;THESE TWO INSTRUCTIONS GETS
1100 037140  010077  142070                     MOV     R0,@RHBA        ;ADDR. OF WRFROM INTO R0 AND
1101                                                                    ;BUS ADDRESS REGISTER
1102 037144  012720  010000                     MOV     #FMT22,(R0)+    ;FORMAT=16 BIT WORDS
1103                                                                    ;CYLINDER=0
1104 037150  012720  000026                     MOV     #22.,(R0)+      ;TRACK=0, SECTOR=22, KEYS=0
1105 037154  005020                     CLR     (R0)+           ;KEY1=0
1106 037156  005020                     CLR     (R0)+           ;KEY2=0
1107 037160  012705  000400                     MOV     #256.,R5        ;COUNTER
1108 037164  012720  177777             1$:     MOV     #-1,(R0)+       ;MOVE ALL ONES FOR DATA
1109 037170  005305                     DEC     R5
1110 037172  001374                     BNE     1$                      ;BRANCH IF DATA NOT COMPLETE
1111 037174  012777  000026  142042            MOV     #22.,@RHDST     ;TRACK=0 SECTOR=22
1112
1113 037202  004737  041514                     JSR     PC,CHECKT       ;CHECK THAT DVA,RDY,DPR,DRY = 1
     037206  104401  062726                     TYPE    ,CPHALT         ;AND THAT NO OTHERS = 1.  CANNOT CON-
                                                                        ;TINUE TESTING IF BOTH AREN'T TRUE
     037212  000000                     HALT                            ;STOP THE TEST
1114
1115 037214  013711  001464                     MOV     WRIFOR,@R1      ;GET READY FOR WRITE HEADER AND
1116                                                                    ;DATA WITH 62 IN RHCS1
1117 037220  005037  001406                     CLR     ERFLG$  ;CLEAR ERROR FLAG
1118 037224  012777  010000  142016            MOV     #FMT22,@RHOF    ;FORMA BIT=1 (16 BIT WORDS)
1119 037232  005077  142014                     CLR     @RHCA           ;CYLINDER =0
1120
1121                                    ;AS EXCEPTION IS ASSERTED BEFORE RUN IS
1122                                    ;LATCHED RHWC,RHBA,RHCS1,RHCS2 CANNOT BE CHECKED
1123                                    ;BECAUSE RHWC WILL VARY DEPENDING UPON GATE DELAYS
1124                                    ;ON DIFFERENT UNITS
1125
1126                                    ;THE REGISTERS WILL BE SAVED IN REINTO BUFFER
1127 037236  004037  042152                     JSR     R0,SAVER        ;SAVE
1128 037242  001242                     RHER1                           ;FROM
1129 037244  002554                     REINTO                          ;TO
1130 037246  000015                     13.                             ;NUMBER SAVED
1131
1132                                    ;GO TO WRITE HEADER AND DATA
1133
1134 037250  013700  001260                     MOV     RHMR,R0 ;NOW R0 HAS MAINTENANCE REG. ADDR.
1135 037254  012710  000001                     MOV     #DMD,@R0        ;SET DIAGNOSTIC MODE
1136 037260  052777  000001  141752            BIS     #GO,@RHCS1      ;GO
```

```
1137
1138                                    ;CHANGE SAVED REGISTERS TO EXPECTED VALUE
1139 037266  012737  002000  002554            MOV     #IAE,REINTO     ;SAVED RHER1
1140 037274  012737  000027  002556            MOV     #23.,REINTO+2;SAVED RHDST
1141 037302  013737  001416  002570            MOV     ATTENT,REINTO+14 ;SAVED RHAS
1142 037310  052737  000001  002572            BIS     #DMD,REINTO+16 ;SAVED RHMR
1143 037316  052737  140000  002574            BIS     #ATA!ERR,REINTO+20 ;SAVED RHDS1
1144
1145                                    ;SAVE REGISTERS AGAIN SO THAT COMPARES CAN BE DONE
1146 037324  004037  042152                    JSR     R0,SAVER        ;SAVE
1147 037330  001242                            RHER1                   ;FROM
1148 037332  001510                            WRFROM                  ;TO
1149 037334  000015                            13.                     ;NUMBER OF REGISTERS SAVED
1150
1151                                    ;AS UPPER BYTE OF RHAS CAN BE CHANGING IN A DUAL PORT
1152                                    ;OPERATION THE UPPER BYTE OF RHAS WILL BE SAVED AS IS
1153                                    ;SO THAT THE COMPARES ARE ONLY VALID FOR THE LOWER BYTE
1154 037336  113737  002571  001525            MOVB    REINTO+15,WRFROM+15;SAVE UPPER RHAS
1155
1156
1157                                    ;COMPARE REGISTERS BEFORE READ IN PRESET COMMAND
1158                                    ;WITH AFTER COMMAND
1159 037344  004037  042354                    JSR     R0,COMPAR       ;COMPARE
1160 037350  002554                            REINTO                  ;GOOD BUFFER
1161 037352  001510                            WRFROM                  ;TEST BUFFER
1162 037354  000015                            13.                     ;NUMBER OF REGISTERS
1163 037356  037364                            2$                      ;RETURN FOR ERROR
1164 037360  037364                            2$                      ;SAME
1165 037362  037404                            3$                      ;RETURN FOR GOCD COMPARISON
1166
1167 037364  013705  046426            2$:     MOV     ERWORD,R5       ;GETTING READY TO INDEX
1168 037370  060505                            ADD     R5,R5           ;DOUBLE ERROR WORD
1169 037372  016537  001240  041130            MOV     RHER1-2(R5),REGADR ;FAILING REG. ADDRESS
1170 037400  104001                            EMT     1
1171                                                                    ;REGISTER CHANGE
1172 037402  000207                            RTS     PC              ;RETURN FOR FURTHER COMPARISONS
1173
1174                                                                    ;NO ERRORS
1175
1176 037404  004737  041460            3$:     JSR     PC,CLDISK       ;CLEAR GO BIT
1177
1187                                    ;;*****************************************************************
                                        ;*TEST 122      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
                                        ;;*****************************************************************
     037410  000004                     TST122: SCOPE
1188 037412  012706  001100                    MOV     #STACK,SP       ;RESET STACK
1189 037416  012737  000122  001432            MOV     #122,TSTNM      ;MOVE #122 TO TEST NUMBER
1190 037424  004737  041460                    JSR     PC,CLDISK       ;CLEAR REGISTERS AND SET UNIT NO.
1191
1192                                    ;GIVE INDEX PULSE
```

```
1193 037430  012777  000001  141622          MOV     #DMD,@RHMR      ;SET DIAGNOSTIC MODE
1194 037436  052777  000004  141614          BIS     #MINX,@RHMR     ;SET INDEX
1195 037444  042777  000004  141606          BIC     #MINX,@RHMR     ;CLEAR INDEX
1196
1197                                 ;THESE ARE REGULAR SETUPS
1198
1199 037452  012777  177400  141552          MOV     #-256.,@RHWC    ;256 DATA WORDS 4 HEADER WORDS
1200 037460  012700  001510                  MOV     #WRFROM,R0      ;THESE TWO INSTRUCTIONS GETS
1201 037464  010077  141544                  MOV     R0,@RHBA        ;ADDR. OF WRFROM INTO R0 AND
1202                                                                 ;BUS ADDRESS REGISTER
1203 037470  012705  000400                  MOV     #256.,R5        ;COUNTER
1204 037474  012720  177777          1$:     MOV     #-1,(R0)+       ;MOVE ALL ONES FOR DATA
1205 037500  005305                          DEC     R5
1206 037502  001374                          BNE     1$              ;BRANCH IF DATA NOT COMPLETE
1207 037504  012777  000000  141532          MOV     #0.,@RHDST      ;TRACK=0 SECTOR=0
1208
1209 037512  004737  041514                  JSR     PC,CHECKT       ;CHECK THAT DVA,RDY,DPR,DRY = 1
     037516  104401  062726                  TYPE    ,CPHALT         ;AND THAT NO OTHERS = 1.  CANNOT CON-
                                                                     ;TINUE TESTING IF BOTH AREN'T TRUE
     037522  000000                          HALT                    ;STOP THE TEST
1210
1211 037524  013711  001462                  MOV     WRIDAT,@R1      ;GET READY FOR WRITE
1212                                                                 ;DATA WITH 60 IN RHCS1
1213 037530  005037  001406                  CLR     ERFLG$  ;CLEAR ERROR FLAG
1214 037534  012777  010000  141506          MOV     #FMT22,@RHOF    ;FORMA BIT=1 (16 BIT WORDS)
1215
1216 037542  005737  001434                  TST     RP06    ;MOVE FLAG TO ITSELF TO TEST
1217 037546  001404                          BEQ     4$              ;TREAT DRIVE AS RP04 IF FLAG = 0
1218
1219 037550  012777  001457  141474          MOV     #815.,@RHCA     ;CYLINDER = 815 (ONE TOO MANY)
1220 037556  000403                          BR      5$              ;TREAT DRIVE AS RP06
1221
1222 037560  012777  000633  141464  4$:     MOV     #411.,@RHCA     ;CYLINDER = 411 (ONE TOO MANY)
1223                                                                 ;TREAT DRIVE AS RP04
1224
1225                                 ;AS EXCEPTION IS ASSERTED BEFORE RUN IS
1226                                 ;LATCHED RHWC,RHBA,RHCS1,RHCS2 CANNOT BE CHECKED
1227                                 ;BECAUSE RHWC WILL VARY DEPENDING UPON GATE DELAYS
1228                                 ;ON DIFFERENT UNITS
1229
1230                                 ;THE REGISTERS WILL BE SAVED IN REINTO BUFFER
1231 037566  004037  042152          5$:     JSR     R0,SAVER        ;SAVE
1232 037572  001242                          RHER1                   ;FROM
1233 037574  002554                          REINTO                  ;TO
1234 037576  000015                          13.                     ;NUMBER SAVED
1235
1236                                 ;GO TO WRITE HEADER AND DATA
1237
1238 037600  013700  001260                  MOV     RHMR,R0 ;NOW R0 HAS MAINTENANCE REG. ADDR.
1239 037604  012710  000001                  MOV     #DMD,@R0        ;SET DIAGNOSTIC MODE
1240 037610  052777  000001  141422          BIS     #GO,@RHCS1      ;GO
1241
1242                                 ;CHANGE SAVED REGISTERS TO EXPECTED VALUE
1243 037616  012737  002000  002554          MOV     #IAE,REINTO     ;SAVED RHER1
1244 037624  012737  000001  002556          MOV     #1.,REINTO+2;SAVED RHDST
1245 037632  013737  001416  002570          MOV     ATTENT,REINTO+14 ;SAVED RHAS
1246 037640  052737  000001  002572          BIS     #DMD,REINTO+16 ;SAVED RHMR
```

```
1247 037646  052737  140000  002574          BIS     #ATA!ERR,REINTO+20 ;SAVED RHDS1
1248
1249                                 ;SAVE REGISTERS AGAIN SO THAT COMPARES CAN BE DONE
1250 037654  004037  042152                  JSR     R0,SAVER          ;SAVE
1251 037660  001242                          RHER1                     ;FROM
1252 037662  001510                          WRFROM                    ;TO
1253 037664  000015                          13.                       ;NUMBER OF REGISTERS SAVED
1254
1255                                 ;AS UPPER BYTE OF RHAS CAN BE CHANGING IN A DUAL PORT
1256                                 ;OPERATION THE UPPER BYTE OF RHAS WILL BE SAVED AS IS
1257                                 ;SO THAT THE COMPARES ARE ONLY VALID FOR THE LOWER BYTE
1258 037666  113737  002571  001525          MOVB    REINTO+15,WRFROM+15;SAVE UPPER RHAS
1259
1260
1261                                 ;COMPARE REGISTERS BEFORE READ IN PRESET COMMAND
1262                                 ;WITH AFTER COMMAND
1263 037674  004037  042354                  JSR     R0,COMPAR         ;COMPARE
1264 037700  002554                          REINTO                    ;GOOD BUFFER
1265 037702  001510                          WRFROM                    ;TEST BUFFER
1266 037704  000015                          13.                       ;NUMBER OF REGISTERS
1267 037706  037714                          2$                        ;RETURN FOR ERROR
1268 037710  037714                          2$                        ;SAME
1269 037712  037734                          3$                        ;RETURN FOR GOOD COMPARISON
1270
1271 037714  013705  046426          2$:     MOV     ERWORD,R5         ;GETTING READY TO INDEX
1272 037720  060505                          ADD     R5,R5             ;DOUBLE ERROR WORD
1273 037722  016537  001240  041130          MOV     RHER1-2(R5),REGADR ;FAILING REG. ADDRESS
1274 037730  104001                          EMT     1
1275                                                                   ;REGISTER CHANGE
1276 037732  000207                          RTS     PC                ;RETURN FOR FURTHER COMPARISONS
1277
1278                                                                   ;NO ERRORS
1279
1280 037734  004737  041460          3$:     JSR     PC,CLDISK         ;CLEAR GO BIT
1281
1288                                 ;;*****************************************************************
                                     ;*TEST 123       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
                                     ;;*****************************************************************
     037740  000004                  TST123: SCOPE
     037742  012737  000001  001212          MOV     #1,$TIMES         ;;DO 1 ITERATION
1289 037750  012737  000000  177776          MOV     #0,PS             ;REINSTATE PS TO 0
1290 037756  104401  037764                  TYPE    ,65$              ;;TYPE ASCIZ STRING
     037762  000425                          BR      64$               ;;GET OVER THE ASCIZ
                                     ;;65$:  .ASCIZ  <CRLF>/TOTAL ERRORS FOR THIS PASS ON UNIT NO. /
     040036                          64$:
1291 040036  013746  001374                  MOV     UNIT,-(SP)        ;GET READY TO TYPE UNIT NUMBER
1292 040042  104403                          TYPDS
1293 040044  104401  040052                  TYPE    ,67$              ;;TYPE ASCIZ STRING
     040050  000402                          BR      66$               ;;GET OVER THE ASCIZ
                                     ;;67$:  .ASCIZ  /= /
     040056                          66$:
1294 040056  013746  001112                  MOV     $ERTTL,-(SP)      ;GET READY TO TYPE NUMBER OF ERRORS
1295 040062  104405                          TYPDS
```

```
1296 040064  005037  001112                CLR     $ERTTL          ;CLEAR TOTAL NUMBER OF ERRORS
1297 040070  005037  001102                CLR     $TSTNM          ;CLEAR TEST NUMBER
1298 040074  005737  001402                TST     SELECT          ;STARTING FROM 210 ?
1299
1300 040100  001413                       BEQ     3$              ;TEST NEXT DRIVE IF NOT
1301                                                               ;CONTINUE ON THIS ONE IF SO
1302 040102  005237  001100                INC     $PASS           ;INCREASE PASS COUNT
1303 040106  104401  040271                TYPE    ,$ENDMG         ;TYPE END PASS #
1304 040112  013746  001100                MOV     $PASS,-(SP)
1305 040116  104405                        TYPDS
1306 040120  104401  040266                TYPE    ,$ENULL
1307 040124  000137  006440                JMP     TST5            ;DO NEXT TESTS
1308
1309 040130  005337  001376          3$:   DEC     NOUNITS         ;NO. OF UNITS PRESENT DECREMENTED
1310 040134  001413                       BEQ     $EOP            ;BRANCH IF ALL DRIVES COMPLETE
1311 040136  013700  001374                MOV     UNIT,R0         ;UNIT UNDER TEST
1312 040142  012701  001354                MOV     #UNITS,R1       ;TABLE
1313 040146  022100                  1$:   CMP     (R1)+,R0        ;IS THIS UNIT JUST TESTED
1314 040150  001401                       BEQ     2$              ;BRANCH IF YES
1315 040152  000775                       BR      1$              ;BRANCH IF NO
1316 040154  011137  001374          2$:   MOV     (R1),UNIT       ;THIS IS NEXT UNIT
1317 040160  000137  006440                JMP     TST5            ;GO FOR NEXT TESTS
```

```
                                .SBTTL  SUBROUTINES
                                .SBTTL  END OF PASS ROUTINE

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

040164                          $EOP:
040164  000004                         SCOPE
040166  005037  001102                  CLR     $TSTNM          ;;ZERO THE TEST NUMBER
040172  005037  001212                  CLR     $TIMES          ;;ZERO THE NUMBER OF ITERATIONS
040176  005237  001100                  INC     $PASS           ;;INCREMENT THE PASS NUMBER
040202  042737  100000  001100        BIC     #100000,$PASS   ;;DON'T ALLOW A NEG. NUMBER
040210  005327                  DEC     (PC)+           ;;LOOP?
040212  000001                  $EOPCT: .WORD   1
040214  003022                  BGT     $DOAGN          ;;YES
040216  012737                  MOV     (PC)+,@(PC)+    ;;RESTORE COUNTER
040220  000001                  $ENDCT: .WORD   1
040222  040212                          $EOPCT
040224  104401  040271                  TYPE    ,$ENDMG         ;;TYPE 'END PASS #''
040230  013746  001100                  MOV     $PASS,-(SP)     ;;SAVE $PASS FOR TYPEOUT
040234  104405                  TYPDS                   ;;GO TYPE--DECIMAL ASCII WITH SIGN
040236  104401  040266                  TYPE    ,$ENULL         ;;TYPE A NULL CHARACTER
040242  013700  000042          $GET42: MOV     @#42,R0         ;;GET MONITOR ADDRESS
040246  001405                  BEQ     $DOAGN          ;;BRANCH IF NO MONITOR
040250  000005                  RESET                   ;;CLEAR THE WORLD
040252  004710                  $ENDAD: JSR     PC,(R0)         ;;GO TO MONITOR
040254  000240                  NOP                     ;;SAVE ROOM
040256  000240                  NOP                     ;;FOR
040260  000240                  NOP                     ;;ACT11
040262                          $DOAGN:
040262  000137                  JMP     @(PC)+          ;;RETURN
040264  005052                  $RTNAD: .WORD   TST1
040266     377     377     000  $ENULL: .BYTE   -1,-1,0         ;;NULL CHARACTER STRING
040271     015     012     105  $ENDMG: .ASCIZ  <15><12>/END PASS #/
```

```
                                ;;*****************************************************************
                                ;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.
                                ;;*****************************************************************

040306  000000                  TESTAD: 0                       ;FIRST ADDRESS OF TEST

040310                          OPERSEL:
040310  005037  177776                  CLR     PS              ;MAKE PROCESSOR STATUS ZERO
040314  104401  040322                  TYPE    ,65$            ;;TYPE ASCIZ STRING
040320  000421                          BR      64$             ;;GET OVER THE ASCIZ
                                ;;65$:  .ASCIZ  <CRLF>/THE PROGRAM WAS IN TEST NUMBER /
040364                          64$:
040364  013746  001432                  MOV     TSTNM,-(SP)     ;GET READY TO TYPE TEST
040370  104402                          TYPOC                   ;NUMBER
040372  104401  040400                  TYPE    ,67$            ;;TYPE ASCIZ STRING
040376  000414                          BR      66$             ;;GET OVER THE ASCIZ
                                ;;67$:  .ASCIZ  <CRLF>/THE LOOP BACK PC WAS /
040430                          66$:
040430  013746  001110                  MOV     $LPERR,-(SP)    ;GET READY TO TYPE LOOP BACK PC
040434  104402                          TYPOC
040436  104401  001223                  TYPE    ,$CRLF
040442  104401  040450                  TYPE    ,69$            ;;TYPE ASCIZ STRING
040446  000430                          BR      68$             ;;GET OVER THE ASCIZ
                                ;;69$:  .ASCIZ  <CRLF>/SET SWITCH FOR LOOP ON ERROR OR LOOP ON TEST,/
040530                          68$:
040530  104401  040536                  TYPE    ,71$            ;;TYPE ASCIZ STRING
040534  000430                          BR      70$             ;;GET OVER THE ASCIZ
                                ;;71$:  .ASCIZ  <CRLF>/TYPE THE FIRST PC OF THE TEST TO BE LOOPED ON,/
040616                          70$:
040616  104401  040624                  TYPE    ,73$            ;;TYPE ASCIZ STRING
040622  000421                          BR      72$             ;;GET OVER THE ASCIZ
                                ;;73$:  .ASCIZ  <CRLF>/FOLLOWED BY A CARRIAGE RETURN /<CRLF>
040666                          72$:
040666  104412                          RDOCT
040670  062716  000002                  ADD     #2,(SP)         ;GET LPADR
040674  012637  001106                  MOV     (SP)+,$LPADR
040700  104401  040706                  TYPE    ,75$            ;;TYPE ASCIZ STRING
040704  000433                          BR      74$             ;;GET OVER THE ASCIZ
                                ;;75$:  .ASCIZ  <CRLF>/TYPE THE PC WHERE YOU WANT THE PROGRAM TO LOOP BACK/
040774                          74$:
```

END OF PASS ROUTINE

```
     040774  104401  041002                 TYPE    ,77$            ;;TYPE ASCIZ STRING
     041000  000423                         BR      76$             ;;GET OVER THE ASCIZ
                                     ;;77$:  .ASCIZ  <CRLF>/TO, FOLLOWED BY A CARRIAGE RETURN /<CRLF>
     041050                          76$:
41   041050  104412                         RDOCT
42   041052  012637  001110                 MOV     (SP)+,$LPERR    ;GET LPERR
43   041056  013746  001106                 MOV     $LPADR,-(SP)
44   041062  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

041064                          PUTREG:
041064  010046                          MOV     R0,-(SP)        ;;PUSH R0 ON STACK
041066  010146                          MOV     R1,-(SP)        ;;PUSH R1 ON STACK
041070  010246                          MOV     R2,-(SP)        ;;PUSH R2 ON STACK
041072  012700  001232                  MOV     #RHWC,R0        ;STARTING ADDRESS OF REG
041076  012701  001306                  MOV     #WC,R1          ;STARTING ADDRESS OF WERE SAVED
041102  012702  000023                  MOV     #RHCC-RHWC+2/2,R2 ;NUMBER OF REG. INTO R2
041106  013021                  10$:    MOV     @(R0)+,(R1)+    ;SAVE HARDWARE REG.
041110  005302                          DEC     R2
041112  001375                          BNE     10$
041114  012602                          MOV     (SP)+,R2        ;;POP STACK INTO R2
041116  012601                          MOV     (SP)+,R1        ;;POP STACK INTO R1
041120  012600                          MOV     (SP)+,R0        ;;POP STACK INTO R0
041122  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

041124  000000                  MASK:   0                       ;BITS UNDER TEST
041126  000000                  LERR:   0                       ;ERROR HLT ADDRESS
041130  000000                  REGADR: 0

041132  012537  041124          BITST:  MOV     (R5)+,MASK      ;FETCH DATA MASK
041136  012504                          MOV     (R5)+,R4        ;GET ADDRESS OF REG. UNDER TEST
041140  010437  041130                  MOV     R4,REGADR
041144  010537  041126                  MOV     R5,LERR         ;GET ERROR RETURN ADDR.
041150  062705  000004                  ADD     #4,R5           ;MODIFY RETURN ADDR. TO JUMP OVER RTS
041154  012703  000001                  MOV     #1,R3           ;INITIALIZE DATA PATTERN
041160  004737  041202          BLT1:   JSR     PC,BLT2         ;OUTPUT FLOATING ZERO
041164  004737  041202                  JSR     PC,BLT2         ;OUTPUT FLOATING ONE
041170  000241                          CLC
041172  006103                          ROL     R3              ;SHIFT PATTERN
041174  005703                          TST     R3
041176  001370                          BNE     BLT1            ;BRANCH IF NOT COMPLETE
041200  000205                          RTS     R5              ;RETURN TO TEST
041202  005103                  BLT2:   COM     R3              ;COMPLEMENT PATTERN
041204  012737  041212  041440          MOV     #BLT3,LAD       ;SET SCOPE LOOP
041212  032777  001000  137720  BLT3:   BIT     #SW09,@SWR      ;LOOP ON ERROR
041220  001411                          BEQ     4$              ;BRANCH IF NO
041222  105737  001103                  TSTB    $ERFLG  ;ANY ERRORS
041226  001406                          BEQ     4$      ;BRANCH IF NO
041230  000005                          RESET                   ;START WITH AN INIT
041232  013777  001374  137776          MOV     UNIT,@RHCS2     ;SET UNIT NUMBER UNDER TEST
041240  004737  054216                  JSR     PC,$TKINT       ;INITILIZE TK
```

```
                                                                ;INIT FOR SCOPING LOOPS
041244  010337  001124          4$:   MOV     R3,$GDDAT         ;STORE GOOD DATA
041250  005137  041124                COM     MASK              ;AND MASK WITH PATTERN
041254  043737  041124  001124        BIC     MASK,$GDDAT       ;CLEAR THE REST
041262  005137  041124                COM     MASK              ;RESTORE MASK
041266  013714  001124                MOV     $GDDAT,(R4)       ;OUTPUT TO REGISTER
041272  011437  001126                MOV     (R4),$BDDAT       ;INPUT FROM REGISTER
041276  005137  041124                COM     MASK
041302  043737  041124  001126        BIC     MASK,$BDDAT       ;AND MASK OUT RECEIVED DATA
041310  005137  041124                COM     MASK              ;RESTORE MASK
041314  023737  001124  001126        CMP     $GDDAT,$BDDAT     ;IS DATA CORRECT
041322  001424                      BEQ     1$                ;BRANCH IF GOOD
041324  011437  001126                MOV     (R4),$BDDAT
041330  023704  001240                CMP     RHCS1,R4          ;REGISTER UNDER TEST RHCS1?
041334  001004                      BNE     2$                ;BRANCH IF NOT
041336  052737  004200  001124        BIS     #RDY!DVA,$GDDAT;SET RDY AND DVA
041344  000410                      BR      3$
041346  023704  001236          2$:   CMP     RHCS2,R4          ;REGISTER UNDER TEST RHCS2?
041352  001005                      BNE     3$                ;BRANCH IF NOT
041354  011446                      MOV     @R4,-(SP)         ;GET RHCS2
041356  042716  177477              BIC     #^C<IR!OR>,(SP)   ;KEEP IR AND OR BIT
041362  052637  001124              BIS     (SP)+,$GDDAT      ;SET IR OR BITS IF NEEDED
041366  004777  177534          3$:   JSR     PC,@LERR          ;GO TO REPORT ERROR
041372  000240                      NOP                       ;REPLACE BY 104420 FOR LOCAL SCOPE LOOP
041374  000207          1$:   RTS     PC
```

CLEAR MEMORY ROUTINE

```
                                .SBTTL  CLEAR MEMORY ROUTINE

                                ;THIS CLEARS ANY BLOCK OF MEMORY
                                ;FILLING IT WITH ANY DATA
                                ;
                                ;CALL
                                ;       JSR     R0,CLAREA
                                ;       .WORD   X               ;STARTING ADDRESS OF BLOCK
                                ;       .WORD   Y
                                ;       .WORD   Z               ;DATA TO BE FILLED
                                ;
                                ;R1 WILL HAVE STARTING ADDRESS OF BLOCK TO BE FILLED
                                ;R2 AFTER SUBTRACTION WILL HAVE TWICE NUMBER OF LOCATIONS
                                ;R3 WILL HAVE DATA TO BE FILLED
                                ;TO AVOID DIVIDE ROUTINE TWO DECREMENT R2 WILL BE USED

041376                          CLAREA:
041376  010146                         MOV     R1,-(SP)        ;;PUSH R1 ON STACK
041400  010246                         MOV     R2,-(SP)        ;;PUSH R2 ON STACK
041402  010346                         MOV     R3,-(SP)        ;;PUSH R3 ON STACK
041404  012001                         MOV     (R0)+,R1        ;FROM
041406  012002                         MOV     (R0)+,R2        ;TO
041410  012003                         MOV     (R0)+,R3        ;DATA
041412  160102                         SUB     R1,R2           ;NO. OF LOCATIONS MINUS TWO
041414  062702  000002                 ADD     #2,R2           ;GET TWICE NO OF LOCATIONS
041420  010321                 1$:     MOV     R3,(R1)+        ;MOVE IN DATA
041422  005302                         DEC     R2
041424  005302                         DEC     R2
041426  001374                         BNE     1$              ;BRANCH IF NOT COMPLETE
041430  012603                         MOV     (SP)+,R3        ;;POP STACK INTO R3
041432  012602                         MOV     (SP)+,R2        ;;POP STACK INTO R2
041434  012601                         MOV     (SP)+,R1        ;;POP STACK INTO R1
041436  000200                         RTS     R0              ;RETURN
```

```
                                .SBTTL  LOCAL TRAPS
                        ;
041440  000000          LAD:    0
                        ;
041442  032777  001000  137470  T.SCOP: BIT     #SW09,@SWR
041450  001402                          BEQ     1$
041452  013716  041440                  MOV     LAD,(SP)
041456  000002          1$:     RTI
                        ;
                        ;EXAMPLE OF THE USE OF THE ABOVE
                        ;THIS WILL LOOP BETWEEN X: AND SCOP1 PROVIDED THERE IS NO 'NEWTST"
                        ;CALL
                        ;       MOV     #X,LAD
                        ;X:     ---     ---
                        ;       ---     ---
                        ;       ---     ---
                        ;       SCOP1
                        ;
                                .SBTTL  CLEAR DISK ROUTINE
                        ;
041460  013701  001240  CLDISK: MOV      RHCS1,R1        ;R1 WILL BE CONTROL AND STATUS1
041464  013702  001236          MOV     RHCS2,R2        ;R2 WILL BE CONTROL AND STATUS2
041470  013703  001262          MOV     RHDS1,R3        ;R3 WILL BE DISK STATUS REGISTER1
041474  013704  001242          MOV     RHER1,R4        ;R4 WILL BE ERROR REGISTER #1
                        ;
041500  012712  000040          MOV     #CLR,@R2        ;CLEAR ALL REG.
041504  013712  001374          MOV     UNIT,@R2        ;REINSTATE UNIT NO.
041510  005011                  CLR     @R1             ;CLEAR FUNCTION BITS
041512  000207                  RTS     PC
```

```
                                .SBTTL  CHECK DISK STATUS ROUTINES

                                ;THIS CHECKS THAT DEVICE AVAILABLE (DVA) AND READY (RDY) IN RHCS1 = 1
                                ;AND CHECKS THAT DEVICE PRESENT (DPR)  DEVICE READY (DRY) IN RHDS1 = 1
                                ;IT ALSO CHECKS THAT NO OTHER BITS IN THESE REGISTERS = 1

041514  011637  001414          CHECKT: MOV     (SP),PCJSR      ;SAVE PC OF JSR+4
041520  162737  000004  001414          SUB     #4,PCJSR        ;GET PC OF JSR
041526  004737  041064                  JSR     PC,PUTREG       ;SAVE REGISTERS
041532  022737  004200  001314          CMP     #DVA!RDY,CS1    ;RHCS1 SHOULD HAVE DEVICE AVAILABLE
                                                                ;AND BE READY
041540  001423                          BEQ     3$              ;BRANCH IF GOOD

041542  032737  004000  001314          BIT     #DVA,CS1        ;BAD SO TEST DEVICE AVAILABLE
041550  001004                          BNE     1$              ;BRANCH IF DVA THERE
041552  010137  001122                  MOV     R1,$BDADR       ;ADDRESS OF BAD REGISTER (RHCS1)
041556  104026                          EMT     26
                                                                ;AVAILABLE AT START OF TEST
041560  000413                          BR      3$              ;BRANCH TO NEXT COMPARE
041562  032737  000200  001314  1$:     BIT     #RDY,CS1        ;TEST READY
041570  001003                          BNE     2$              ;IF RDY THERE BRANCH
041572  010137  001122                  MOV     R1,$BDADR       ;ADDRESS OF BAD REGISTER (RHCS1)
041576  104026                          EMT     26

                                                                ;RIGHT AT START OF TEST
041600  000403                  2$:     BR      3$              ;BRANCH TO NEXT COMPARE
041602  010137  001122                  MOV     R1,$BDADR       ;ADDRESS OF BAD REGISTER (RHCS1)
041606  104026                          EMT     26

                                                                ;THAN DVA AND RDY SET
                                                                ;ALL OTHER BITS SHOULD BE 0

041610  013746  001336          3$:     MOV     DS1,-(SP)       ;GET RHDS1
041614  042716  001100                  BIC     #VV!PROG,(SP)   ;CLEAR VV AND PROGRAMABLE BIT
041620  022726  000600                  CMP     #DPR!DRY,(SP)+;RHDS1 SHOULD HAVE THESE SET
041624  001424                          BEQ     8$              ;BRANCH IF THEY ARE

041626  032737  000400  001336  4$:     BIT     #DPR,DS1        ;TEST DRIVE PRESENT
041634  001004                          BNE     5$              ;CONTINUE IF THERE
041636  010337  001122                  MOV     R3,$BDADR       ;ADDRESS OF BAD REGISTER (RHDS1)
041642  104026                          EMT     26
041644  000413                          BR      7$              ;BRANCH OUT
041646  032737  000200  001336  5$:     BIT     #DRY,DS1        ;TEST DRIVE READY
041654  001004                          BNE     6$              ;IF DPR WAS THERE, BRANCH IF GOOD
041656  010337  001122                  MOV     R3,$BDADR       ;ADDRESS OF BAD REGISTER (RHDS1)
041662  104026                          EMT     26
041664  000403                          BR      7$              ;BRANCH OUT
041666  010337  001122          6$:     MOV     R3,$BDADR       ;ADDRESS OF BAD REGISTER (RHDS1)
041672  104026                          EMT     26

                                                                ;THAN MOL, DRY, DPR, SET
                                                                ;ALL OTHER BITS SHOULD BE 0
041674  000207                  7$:     RTS     PC              ;RETURN TO TEST AND HALT

041676  062716  000006          8$:     ADD     #6,(SP)         ;ADJUST STACK TO GET OVER HALT IN TEST
041702  000207                          RTS     PC              ;RETURN TO TEST AND CONTINUE TESTING
```

```
                                        ;THIS CHECKS THAT DEVICE AVAILABLE (DVA) AND READY (RDY) IN RHCS1 = 1
                                        ;AND CHECKS THAT DEVICE PRESENT (DPR), DEVICE READY (DRY) IN RHDS1 = 1

041704  011637  001414          CHECKE: MOV     (SP),PCJSR      ;SAVE PC OF JSR+4
041710  162737  000004  001414          SUB     #4,PCJSR        ;GET PC OF JSR
041716  004737  041064                  JSR     PC,PUTREG       ;READ & SAVE REGISTERS
041722  032737  000200  001314          BIT     #RDY,CS1        ;RHCS1 SHOULD HAVE DEVICE AVAILABLE
                                                                ;AND BE READY
041730  001004                          BNE     1$              ;BRANCH IF GOOD
041732  010137  001122                  MOV     R1,$BDADR       ;FAILING REGISTER
041736  104026                          EMT     26
                                                                ;DOES NOT HAVE DVA, RDY
041740  000427                          BR      4$              ;BRANCH OUT
041742  032737  004000  001314  1$:     BIT     #DVA,CS1        ;RHCS1 SHOULD HAVE DEVICE AVAILABLE
                                                                ;AND BE READY
041750  001004                          BNE     2$              ;BRANCH IF GOOD
041752  010137  001122                  MOV     R1,$BDADR       ;FAILING REGISTER
041756  104026                          EMT     26
                                                                ;DOES NOT HAVE DVA, RDY
041760  000417                          BR      4$              ;BRANCH OUT
041762  032737  000200  001336  2$:     BIT     #DRY,DS1        ;RHDS1 SHOULD HAVE DPR,DRY
041770  001004                          BNE     3$              ;BRANCH IF THERE
041772  010337  001122                  MOV     R3,$BDADR       ;FAILING REGISTER RHDS1
041776  104026                          EMT     26
042000  000407                          BR      4$              ;BRANCH OUT
042002  032737  000400  001336  3$:     BIT     #DPR,DS1        ;RHDS1 SHOULD HAVE DPR,DRY
042010  001004                          BNE     5$              ;BRANCH OUT AND CONTINUE IF THERE
042012  010337  001122                  MOV     R3,$BDADR       ;FAILING REGISTER RHDS1
042016  104026                          EMT     26

042020  000207          4$:     RTS     PC              ;RETURN TO TEST AND HALT

042022  062716  000006  5$:     ADD     #6,(SP)         ;ADJUST STACK TO GET OVER HALT IN TEST
042026  000207                  RTS     PC              ;RETURN TO TEST AND CONTINUE TESTING
```

```
                                .SBTTL  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)

042030  177777                TIMCNT: 177777                  ;WAITING COUNT
042032  010046                WAIT.T: MOV     R0,-(SP)                ;SAVE R0
042034  016600  000002                MOV     2(SP),R0                ;GET ADRESS OF REG. ADDERSS
042040  010037  001204                MOV     R0,$TMP3                ;WAT PC+2 IN $TMP3
042044  162737  000002  001204        SUB     #2,$TMP3                ;WAT PC FOR TYPEOUT
042052  012037  001176                MOV     (R0)+,$TMP0             ;WAIT REGISTER ADDRESS
042056  012037  001200                MOV     (R0)+,$TMP1             ;WAIT ON BIT
042062  010066  000002                MOV     R0,2(SP)                ;RESTORE RETURN ON STACK
042066  012600                        MOV     (SP)+,R0                ;RESTORE R0
042070  013737  042030  001202        MOV     TIMCNT,$TMP2    ;TEMPORARY COUNT
042076  033777  001200  137072  1$:   BIT     $TMP1,@$TMP0    ;IS REQUIRED BIT THERE?
042104  001021                        BNE     2$              ;BRANCH IF YES
042106  005337  001202                DEC     $TMP2           ;COUNT
042112  001371                        BNE     1$              ;BRANCH IF NOT TIME UP
042114  013737  042030  001202        MOV     TIMCNT,$TMP2    ;TEMPORARY COUNT
042122  033777  001200  137046  3$:   BIT     $TMP1,@$TMP0    ;IS REQUIRED BIT THERE?
042130  001007                        BNE     2$              ;BRANCH IF YES
042132  005337  001202                DEC     $TMP2           ;COUNT
042136  001371                        BNE     3$              ;BRANCH IF NOT TIME UP
042140  017737  137032  001126        MOV     @$TMP0,$BDDAT   ;REGISTER CONTENTS
042146  104016                        EMT     16
042150  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

042152                          SAVER:
042152  010146                          MOV     R1,-(SP)        ;;PUSH R1 ON STACK
042154  010246                          MOV     R2,-(SP)        ;;PUSH R2 ON STACK
042156  010346                          MOV     R3,-(SP)        ;;PUSH R3 ON STACK
042160  012001                          MOV     (R0)+,R1        ;FROM
042162  012002                          MOV     (R0)+,R2        ;TO
042164  012003                          MOV     (R0)+,R3        ;NUMBER
042166  013122                  1$:     MOV     @(R1)+,(R2)+    ;SAVE REGISTER CONTENTS
042170  005303                          DEC     R3              ;COUNT
042172  001375                          BNE     1$              ;BRANCH IF NOT DONE
042174  012603                          MOV     (SP)+,R3        ;;POP STACK INTO R3
042176  012602                          MOV     (SP)+,R2        ;;POP STACK INTO R2
042200  012601                          MOV     (SP)+,R1        ;;POP STACK INTO R1
042202  000200                          RTS     R0


                                        .SBTTL  WRITE CHECK ROUTINE

                                ;THIS IS A SUBROUTINE TO DO WRITE CHECK HEADER AND DATA
                                ;CYLINDER 0, TRACK 1, SECTOR 1, KEYS 0

                                ;THESE ARE TO SET UP FOR DISKLESS USE ONLY
042204  012737  010000  046306  WRCHHD: MOV     #FMT22,CYL      ;CYLINDER 0 FORMAT 16 BIT WORDS
042212  112737  000001  046311          MOVB    #1,SECOTR+1     ;TRACK=1
042220  112737  000001  046310          MOVB    #1,SECOTR       ;SECTOR=1
042226  005037  046312                  CLR     KEY1            ;KEY1=0
042232  005037  046314                  CLR     KEY2            ;KEY2=0
042236  012737  000044  046366          MOV     #36.,DAWORD     ;NO OF DATA WORDS
042244  005037  046316                  CLR     X               ;THIS IS A READ OPERATION
042250  004537  042666                  JSR     R5,CRC  ;GO TO CALCULATE CRC
042254  046306                          CYL
042256  050206                          WCRC

                                ;THESE ARE REGULAR SETUPS

042260  004737  041460                  JSR     PC,CLDISK       ;SET UP GENERAL REGISTERS
                                                                ;AND CLEAR DISK REGISTERS
042264  012777  177730  136740          MOV     #-40.,@RHWC     ;36 DATA WORDS 4 HEADER WORDS
042272  012777  002554  136734          MOV     #REINTO,@RHBA   ;STARTING ADDRESS OF READ BUFFER
042300  112746  000001                  MOVB    #1,-(SP)        ;SECTOR=1
042304  112766  000001  000001          MOVB    #1,1(SP)        ;TRACK=1 IN UPPER BYTE
042312  012677  136726                  MOV     (SP)+,@RHDST    ;TRACK=1, SECTOR=1 IN RHDST
042316  012777  014000  136724          MOV     #FMT22!ECI,@RHOF ;16 BIT WORDS
                                                                ;ECC CORRECTION INHIBIT BECAUSE
                                                                ;ECC LOGIC IS NOT CHECKED YET
```

WRITE CHECK ROUTINE

```
     053 042324  005077  136722          CLR     @RHCA           ;CYLINDER=0
     054 042330  004737  041514          JSR     PC,CHECKT       ;CHECK THAT DVA,RDY,DPR,DRY = 1
         042334  104401  062726          TYPE    ,CPHALT         ;AND THAT NO OTHERS = 1.  CANNOT CON-
                                                                 ;TINUE TESTING IF BOTH AREN'T TRUE
         042340  000000                  HALT                    ;STOP THE TEST
     055 042342  013711  001460          MOV     WRCHDT,@R1      ;WRITE CHECK HEADER AND DATA=52
     056                                                         ;INTO RHCS1
     057 042346  004737  046146          JSR     PC,COMHD        ;WRITE CHECK HEADER AND DATA
     058                                                         ;SAME AS READ HEADER AND DATA
     059
     060 042352  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

042354                  COMPAR:
042354  010146                  MOV     R1,-(SP)        ;;PUSH R1 ON STACK
042356  010246                  MOV     R2,-(SP)        ;;PUSH R2 ON STACK
042360  010346                  MOV     R3,-(SP)        ;;PUSH R3 ON STACK
042362  010446                  MOV     R4,-(SP)        ;;PUSH R4 ON STACK
042364  010546                  MOV     R5,-(SP)        ;;PUSH R5 ON STACK
042366  012001                  MOV     (R0)+,R1        ;ADDRESS OF GOOD DATA BUFFER
042370  012002                  MOV     (R0)+,R2        ;ADDRESS OF TEST DATA BUFFER
042372  012003                  MOV     (R0)+,R3        ;NO OF WORDS TO BE COMPARED
042374  012037  001176          MOV     (R0)+,$TMP0     ;RETURN ON ERROR TO PRINT HEADER
042400  012037  001200          MOV     (R0)+,$TMP1     ;RETURN ON ERROR TO PRINT DATA
042404  011000                  MOV     (R0),R0         ;RETURN ON NO ERROR
042406  010304                  MOV     R3,R4           ;NO OF WORDS TO BE COMPARED
042410  005204                  INC     R4
042412  010437  046426  1$:     MOV     R4,ERWORD       ;FOR ERROR WORD NO
042416  022122                  CMP     (R1)+,(R2)+     ;COMPARE GOOD WITH TEST DATA
042420  001426                  BEQ     3$              ;BRANCH IF GOOD

042422  014137  001124          MOV     -(R1),$GDDAT    ;GOOD DATA
042426  014237  001126          MOV     -(R2),$BDDAT    ;BAD DATA
042432  160337  046426          SUB     R3,ERWORD       ;ERROR WORD NO.
042436  005737  001406          TST     ERFLG$  ;ANY ERRORS ALREAY THERE
042442  001003                  BNE     2$              ;BRANCH IF YES
042444  004777  136526          JSR     PC,@$TMP0       ;RETURN TO PRINT HEADER
042450  000402                  BR      5$              ;BRANCH TO AVOID PRINTING NEXT ERROR
042452  004777  136522  2$:     JSR     PC,@$TMP1       ;RETURN TO PRINT DATA
042456  022122          5$:     CMP     (R1)+,(R2)+     ;UNDO -(R1) AND -(R2) FOR ERRORS
042460  017746  136454          MOV     @SWR,-(SP)      ;GET SWITCH SETTING
042464  042716  177177          BIC     #^C600,(SP)     ;KEEP ONLY SWITCH 7 AND 8
042470  022726  000200          CMP     #SW07,(SP)+     ;IS 7 SET AND 8 RESET
042474  001402                  BEQ     4$              ;BRANCH OUT IF YES
042476  005303          3$:     DEC     R3              ;COUNT
042500  001344                  BNE     1$              ;BRANCH IF ALL NOT DEVICE
042502                  4$:
042502  012605                  MOV     (SP)+,R5        ;;POP STACK INTO R5
042504  012604                  MOV     (SP)+,R4        ;;POP STACK INTO R4
042506  012603                  MOV     (SP)+,R3        ;;POP STACK INTO R3
042510  012602                  MOV     (SP)+,R2        ;;POP STACK INTO R2
042512  012601                  MOV     (SP)+,R1        ;;POP STACK INTO R1
042514  000200                  RTS     R0              ;RETURN TO MAIN PROGRAM



                        ;THIS IS A SUBROUTINE TO DO WRITE CHECK DATA
                        ;CYLINDER 0, TRACK 1, SECTOR 1, KEYS 0
```

```
                                        ;THESE ARE TO SET UP FOR DISKLESS USE ONLY
                                        ;
042516  012737  010000  046306  WRCHDA: MOV     #FMT22,CYL      ;CYLINDER 0 FORMAT 16 BIT WORDS
042524  112737  000001  046311          MOVB    #1,SECOTR+1     ;TRACK=1
042532  112737  000001  046310          MOVB    #1,SECOTR       ;SECTOR=1
042540  005037  046312                  CLR     KEY1            ;KEY1=0
042544  005037  046314                  CLR     KEY2            ;KEY2=0
042550  012737  000040  046366          MOV     #32.,DAWORD     ;NO OF DATA WORDS
042556  005037  046316                  CLR     X               ;THIS IS A READ OPERATION
                                        ;
042562  004537  042666                  JSR     R5,CRC          ;GO TO CALCULATE CRC
042566  046306                          CYL
042570  050206                          WCRC
                                        ;
                                        ;THESE ARE REGULAR SETUPS
                                        ;
042572  004737  041460                  JSR     PC,CLDISK       ;SET UP GENERAL REGISTERS
                                                                ;AND CLEAR DISK REGISTERS
                                        ;
042576  012777  177740  136426          MOV     #-32.,@RHWC     ;36 DATA WORDS 4 HEADER WORDS
042604  012777  002554  136422          MOV     #REINTO,@RHBA   ;STARTING ADDRESS OF READ BUFFER
042612  112746  000001                  MOVB    #1,-(SP)        ;SECTOR=1
042616  112766  000001  000001          MOVB    #1,1(SP)        ;TRACK=1 IN UPPER BYTE
042624  012677  136414                  MOV     (SP)+,@RHDST    ;TRACK=1, SECTOR=1 IN RHDST
042630  012777  014000  136412          MOV     #FMT22!ECI,@RHOF        ;16 BIT WORDS
                                                                ;ECC CORRECTION INHIBIT BECAUSE
                                                                ;ECC LOGIC IS NOT CHECKED YET
042636  005077  136410                  CLR     @RHCA           ;CYLINDER=0
042642  004737  041514                  JSR     PC,CHECKT       ;CHECK THAT DVA,RDY,DPR,DRY = 1
042646  104401  062726                  TYPE    ,CPHALT         ;AND THAT NO OTHERS = 1.  CANNOT CON-
                                                                ;TINUE TESTING IF BOTH AREN'T TRUE
042652  000000                          HALT                    ;STOP THE TEST
042654  013711  001456          MOV     WRCHEK,@R1              ;WRITE CHECK DATA=50 INTO RHCS1
042660  004737  046146          JSR     PC,COMHD                ;WRITE CHECK HEADER AND DATA
                                                                ;SAME AS READ HEADER AND DATA
                                        ;
042664  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

042666                          CRC:
042666  010046                          MOV     R0,-(SP)        ;;PUSH R0 ON STACK
042670  012500                          MOV     (R5)+,R0        ;GET POINTER TO CYL NO.
042672  010146                          MOV     R1,-(SP)        ;;PUSH R1 ON STACK
042674  010246                          MOV     R2,-(SP)        ;;PUSH R2 ON STACK
042676  010346                          MOV     R3,-(SP)        ;;PUSH R3 ON STACK
042700  010446                          MOV     R4,-(SP)        ;;PUSH R4 ON STACK
042702  005001                          CLR     R1              ;CLEAR WORKING LOCATION
042704  005037  001210                  CLR     $TMP5
042710  012737  000004  001176          MOV     #4,$TMP0        ;WORD COUNT
042716  012037  001204          16$:    MOV     (R0)+,$TMP3     ;TEMPORARY WORD STORAGE
042722  012737  000020  001202          MOV     #16.,$TMP2      ;BIT COUNT
042730  013737  001204  001206          MOV     $TMP3,$TMP4     ;TEMPORARY WORD STORAGE
042736  006037  001204          15$:    ROR     $TMP3           ;GET LSB INTO "C"
042742  006037  001210                  ROR     $TMP5           ;GET ABOVE "C" INTO $TMP5
042746  032701  000001                  BIT     #BIT0,R1        ;IS POSITION 15 HIGH
042752  001403                          BEQ     1$              ;BRANCH IF POSITION 16 LOW
042754  012703  100000                  MOV     #BIT15,R3       ;GET POSITION 16
042760  000401                          BR      2$
042762  005003                  1$:     CLR     R3              ;GET POSITION 16
042764  063703  001210          2$:     ADD     $TMP5,R3        ;XOR POSITION 16 WITH D
                                                                ;TO GIVE B
042770  032701  040000                  BIT     #BIT14,R1       ;IS POSITION 2 HIGH
042774  001403                          BEQ     3$              ;BRANCH IF POSITION 2 LOW
042776  012702  100000                  MOV     #BIT15,R2       ;GET POSITION 2
043002  000401                          BR      4$
043004  005002                  3$:     CLR     R2              ;GET POSITION 2
043006  060302                  4$:     ADD     R3,R2           ;XOR B WITH POSITION 2
                                                                ;TO GIVE C
043010  032701  000002                  BIT     #BIT1,R1        ;IS POSITION 15 HIGH
```

```
 54 043014  001403                      BEQ     5$              ;BRANCH IF POSITION 15 LOW
 55 043016  012704  100000               MOV     #BIT15,R4       ;GET POSITION 15
 56 043022  000401                      BR      6$
 57 043024  005004              5$:     CLR     R4              ;GET POSITION 15
 58 043026  060304              6$:     ADD     R3,R4           ;XOR POSITION 15 WITH B
 59                                                             ;TO GIVE E
 60 043030  006037  001206              ROR     $TMP4           ;GET LSB INTO "C"
 61 043034  006001                      ROR     R1              ;GET ABOVE C INTO R1
 62 043036  005703                      TST     R3              ;TEST B
 63 043040  100403                      BMI     7$              ;BRANCH IF B=1
 64 043042  042701  100000              BIC     #BIT15,R1       ;SET B IN POSITION 1
 65 043046  000402                      BR      10$
 66 043050  052701  100000      7$:     BIS     #BIT15,R1       ;SET B IN POSITION 1
 67 043054  005702              10$:    TST     R2              ;TEST C
 68 043056  100403                      BMI     11$             ;BRANCH IF C=1
 69 043060  042701  020000              BIC     #BIT13,R1       ;GET C IN POSITION 3
 70 043064  000402                      BR      12$
 71 043066  052701  020000      11$:    BIS     #BIT13,R1       ;GET C IN POSITION 3
 72 043072  005704              12$:    TST     R4              ;TEST E
 73 043074  100403                      BMI     13$             ;BRANCH IF E=1
 74 043076  042701  000001              BIC     #BIT0,R1        ;GET E IN POSITION 16
 75 043102  000402                      BR      14$
 76 043104  052701  000001      13$:    BIS     #BIT0,R1        ;GET E IN POSITION 16
 77 043110  005337  001202      14$:    DEC     $TMP2           ;BIT COUNTER
 78 043114  001310                      BNE     15$             ;BRANCH IF 16 NOT DONE
 79 043116  005337  001176              DEC     $TMP0           ;WORD COUNTER
 80 043122  001275                      BNE     16$             ;BRANCH IF 4 NOT DONE
 81 043124  010135                      MOV     R1,@(R5)+       ;PUT CRC WHERE DESIRED
 82 043126  012604                      MOV     (SP)+,R4        ;;POP STACK INTO R4
    043130  012603                      MOV     (SP)+,R3        ;;POP STACK INTO R3
    043132  012602                      MOV     (SP)+,R2        ;;POP STACK INTO R2
    043134  012601                      MOV     (SP)+,R1        ;;POP STACK INTO R1
    043136  012600                      MOV     (SP)+,R0        ;;POP STACK INTO R0
 83 043140  000205                      RTS     R5
 84
 85                             ;THIS IS A SUBROUTINE TO SET UP THE SIMULATOR DISK FOR
 86                             ;CYLINDER 0 (16 BITS PER WORD)
 87                             ;TRACK 1, SECTOR 1
 88                             ;KEY1 1
 89                             ;KEY2 1
 90                             ;CRC THROUGH THE JSR R5,CRC
 91                             ;256 WORDS OF 177400
 92
 93                             ;CALL   JSR     PC,SETDSK
 94
 95 043142                      SETDSK:
    043142  010046                      MOV     R0,-(SP)        ;;PUSH R0 ON STACK
    043144  010146                      MOV     R1,-(SP)        ;;PUSH R1 ON STACK
    043146  010246                      MOV     R2,-(SP)        ;;PUSH R2 ON STACK
 96 043150  012700  177400              MOV     #177400,R0      ;DATA IN THE DISK
 97 043154  012701  000400              MOV     #256.,R1        ;COUNTER
 98 043160  012702  050224              MOV     #DISK,R2        ;START OF SIMULATOR DISK
 99 043164  010022              1$:     MOV     R0,(R2)+        ;MOVE IN DATA
100 043166  005301                      DEC     R1              ;COUNT FOR 256
101 043170  001375                      BNE     1$              ;BRANCH IF 256 NOT COMPLETE
102 043172  012701  000021              MOV     #17.,R1         ;2 ECC WORDS, 1 DATA GAP
103                                                             ;14 TOLERANCE GAP
```

CRC GENERATION ROUTINE

```
104 043176  005022                  2$:     CLR     (R2)+           ;CLEAR ECC,DATA GAP AND
105                                                                  ;TOLERANCE GAP
106 043200  005301                          DEC     R1              ;COUNT
107 043202  001375                          BNE     2$              ;BRANCH IF NOT COMPLETE
108
109                                 ;NOW SET UP FOR DISKLESS USE
110
111 043204  012737  010000  046306          MOV     #FMT22,CYL      ;CYLINDER 0 (16 BIT WORDS)
112 043212  112737  000001  046311          MOVB    #1,SECOTR+1     ;TRACK=1
113 043220  112737  000001  046310          MOVB    #1,SECOTR       ;SECTOR=1
114 043226  012737  000001  046312          MOV     #1,KEY1 ;KEY1=1
115 043234  012737  000001  046314          MOV     #1,KEY2 ;KEY2=1
116 043242  013737  000400  046366          MOV     256.,DAWORD     ;NO. OF DATA WORDS
117 043250  004537  042666                  JSR     R5,CRC  ;GO TO CALCULATE CRC
118 043254  046306                          CYL                     ;FIRST CRC WORD
119 043256  050206                          WCRC                    ;PUT CALCULATED CRC
120 043260  012602                          MOV     (SP)+,R2        ;;POP STACK INTO R2
    043262  012601                          MOV     (SP)+,R1        ;;POP STACK INTO R1
    043264  012600                          MOV     (SP)+,R0        ;;POP STACK INTO R0
121 043266  000207                          RTS     PC
```

CRC GENERATION ROUTINE

```
                                        ;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

043270  010037  001414          HCCRCE: MOV     R0,PCJSR        ;SAVE PC OF JSR+4
043274  162737  000004  001414          SUB     #4,PCJSR        ;GET PC OF JSR
043302  004737  041460                  JSR     PC,CLDISK       ;INIT AND SETUP GENERAL REG.
043306  004737  041514                  JSR     PC,CHECKT       ;CHECK THAT DVA,RDY,DPR,DRY = 1
043312  104401  062726                  TYPE    ,CPHALT         ;AND THAT NO OTHERS = 1.  CANNOT CON-
                                                                ;TINUE TESTING IF BOTH AREN'T TRUE
043316  000000                          HALT                    ;STOP THE TEST
043320  011037  001210                  MOV     (R0),$TMP5      ;SAVE COMMAND
043324  012011                          MOV     (R0)+,@R1       ;COMMAND
043326  012077  135720                  MOV     (R0)+,@RHCA     ;CYLINDER
043332  112046                          MOVB    (R0)+,-(SP)     ;SECTOR
043334  105720                          TSTB    (R0)+           ;UP DATE R0
043336  112066  000001                  MOVB    (R0)+,1(SP)     ;TRACK
043342  105720                          TSTB    (R0)+           ;UPDATE R0
043344  012677  135674                  MOV     (SP)+,@RHDST    ;TRACK SECTOR
043350  012077  135656                  MOV     (R0)+,@RHWC     ;NO. OF DATA WORDS +4 HEADER
                                                                ;IF A READ HEADER AND DATA
043354  012077  135654                  MOV     (R0)+,@RHBA     ;STARTING ADDRESS OF BUFFER
043360  012037  046316                  MOV     (R0)+,X ;X=0 READ HEADER AND DATA
                                                                ;X=1 WRITE DATA
043364  012777  014000  135656          MOV     #FMT22!ECI,@RHOF        ;16 BITS PER WORD
                                                                ;ECC CORRECTION INHIBIT
043372  005037  001406                  CLR     ERFLG$          ;CLEAR ERROR FLAG
043376  004737  046146                  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.

043402  004737  041064                  JSR     PC,PUTREG       ;SAVE REGISTERS
043406  005737  001406                  TST     ERFLG$          ;ANY ERRORS ALREADY THERE
043412  001034                          BNE     10$             ;BRANCH IF YES
043414  005737  046316                  TST     X               ;IS THIS A READ
043420  001015                          BNE     3$              ;IF A WRITE DATA BRANCH
```

```
                                        ;NOW THE READ BUFFER WILL BE CHECKED
                                        ;HEADER SHOULD BE COMPLETELY READ AS WRITTEN
                                        ;NO DATA WORDS SHOULD BE READ
                                        ;REINTO BUFFER HAS BEEN FILLED WITH 0
                                        ;WRFROM BUFFER HAS BEEN FILLED WITH EXPECTED DATA

 043422  004037  042354                       JSR     R0,COMPAR       ;CHECK
 043426  001510                               WRFROM                  ;GOOD DATA
 043430  002554                               REINTO                  ;TEST BUFFER
 043432  000400                               256.                    ;4 HEADER 252 DATA
 043434  043442                               1$                      ;RETURN POINT FOR ERROR HEADER
 043436  043446                               2$                      ;RETURN POINT FOR ERROR DATA
 043440  043504                               10$                     ;RETURN FOR GOOD COMPARISON
 043442                                1$:
 043442  104004                               EMT     4
 043444  000207                               RTS     PC              ;RETURN TO COMPARISON SUBROUTINE
 043446                                2$:
 043446  104005                               EMT     5
                                                                      ;HEADER WORDS AND HENCE
                                                                      ;SHOULD BE READ AS WRITTEN ON
                                                                      ;DISK, WORD NOS. 5 ONWARDS
                                                                      ;SHOULD NOT BE READ AND HENCE
                                                                      ;READ INTO BUFFER
                                                                      ;SHOULD BE UNCHANGED
 043450  000207                               RTS     PC              ;RETURN TO COMPARISON

 043452  000414                               BR      10$             ;JUMP OUT

                                        ;NOW THE DISK WILL BE CHECKED
                                        ;NO DATA SHOULD BE WRITTEN
                                        ;REINTO BUFFER HAS BEEN FILLED WITH EXPECTED DATA
                                        ;DISK HAS BEEN FILLED WITH 177400
                                        ;WRFROM HAS BEEN FILLED WITH 125252

 043454  004037  042354          3$:     JSR     R0,COMPAR       ;CHECK
 043460  002554                          REINTO                  ;GOOD DATA BUFFER
 043462  050224                          DISK                    ;TEST BUFFER
 043464  000400                          256.
 043466  043474                          4$                      ;RETURN POINT FOR ERROR HEADER
 043470  043500                          5$                      ;RETURN POINT FOR ERROR DATA
 043472  043504                          10$                     ;RETURN POINT FOR GOOD COMPARISON
 043474                           4$:
 043474  104004                          EMT     4
 043476  000207                          RTS     PC              ;RETURN TO COMPARISON SUBROUTINE
 043500                           5$:
 043500  104005                          EMT     5
                                                                  ;WORDS THE SHOULD NOT
                                                                  ;HAVE BEEN CHANGED BY THE
                                                                  ;WRITE COMMAND
 043502  000207                          RTS     PC              ;RETURN TO COMPARISON SUBROUTINE
 043504  005720                  10$:    TST     (R0)+           ;IS THIS A HCRC ON HCE CHECK?
 043506  001442                          BEQ     6$              ;BRANCH IF HCRC
 043510  022737  000072  001210          CMP     #72,$TMP5       ;IS THIS A READ COMMAND
 043516  001417                          BEQ     11$             ;BRANCH IF YES
 043520  017737  135516  001126          MOV     @RHER1,$BDDAT   ;TEST DATA
 043526  022737  000200  001126          CMP     #HCE,$BDDAT     ;ONLY HEADER COMPARE BIT?
```

```
                                                        ;SHOULD BE SET
043534  001470                        BEQ     7$      ;BRANCH IF GOOD
043536  013737  001242  041130        MOV     RHER1,REGADR    ;REGISTER ADDRESS RHER1
043544  012737  000200  001124        MOV     #HCE,$GDDAT     ;GOOD DATA
043552  104027                        EMT     27
                                                        ;HEADER ONLY HCE SHOULD
043554  000460                        BR      7$      ;BE SET
043556                          11$:
043556  017737  135460  001126        MOV     @RHER1,$BDDAT   ;TEST DATA
043564  022737  100200  001126        CMP     #DCK!HCE,$BDDAT ;ONLY HEADER COMPARE BIT?
                                                        ;SHOULD BE SET
                                                        ;DCK IS SET BECAUSE ECC IS NOT READ
043572  001451                        BEQ     7$      ;BRANCH IF GOOD
043574  013737  001242  041130        MOV     RHER1,REGADR    ;REGISTER ADDRESS RHER1
043602  012737  100200  001124        MOV     #DCK!HCE,$GDDAT ;GOOD DATA
043610  104027                        EMT     27
                                                        ;HEADER ONLY HCE SHOULD
043612  000441                        BR      7$      ;BE SET
043614  022737  000072  001210  6$:   CMP     #72,$TMP5       ;IS THIS A READ COMMAND?
043622  001417                        BEQ     12$     ;BRANCH IF A READ
043624  017737  135412  001126        MOV     @RHER1,$BDDAT   ;TEST DATA
043632  022737  000400  001126        CMP     #HCRC,$BDDAT    ;ONLY CRC ERROR SHOULD BE THERE
043640  001426                        BEQ     7$
043642  013737  001242  041130        MOV     RHER1,REGADR    ;REG. ADDR = RHER1
043650  012737  000400  001124        MOV     #HCRC,$GDDAT    ;GOOD DATA
043656  104027                        EMT     27
                                                        ;SHOULD BE SET
043660  000416                        BR      7$      ;BRANCH OUT
043662  017737  135354  001126  12$:   MOV     @RHER1,$BDDAT   ;TEST DATA

043670  022737  100400  001126        CMP     #DCK!HCRC,$BDDAT        ;HCRC AND DCK SHOULD BE SET
                                                        ;DCK IS SET BECAUSE ECC IS NOT READ
043676  001407                        BEQ     7$      ;BRANCH IF GOOD
043700  012737  100400  001124        MOV     #DCK!HCRC,$GDDAT        ;GOOD DATA
043706  013737  001242  041130        MOV     RHER1,REGADR    ;FAILING REGISTER RHER1
043714  104027                        EMT     27
                                                        ;DCK AND HCRC SHOULD BE SET
                                                        ;DCK IS SET BECAUSE ECC IS NOT READ
043716  000200                  7$:   RTS     R0      ;RETURN TO MAIN TEST


                              ;THIS IS A SUBROUTINE TO LEAVE AT THE MIDDLE OF
                              ;A WRITE HEADER AND DATA COMMAND
                              ;IT TRYS TO GET SECTOR 10, TRACK 0, CYLINDER 0
                              ;BUT COMES OUT AFTER ONE SECTOR
                              ;THE COMMAND OS JSR PC,MIDDLE
                              ;BAI IS SET

043720                        MIDDLE:
043720  010046                        MOV     R0,-(SP)        ;;PUSH R0 ON STACK
043722  010146                        MOV     R1,-(SP)        ;;PUSH R1 ON STACK
043724  013777  001464  135306        MOV     WRIFOR,@RHCS1   ;WRITE HEADER AND DATA=62
                                                        ;IN RHCS1
043732  012777  177766  135272        MOV     #-10.,@RHWC     ;10 WORDS
043740  012777  001510  135266        MOV     #WRFROM,@RHBA   ;BUS ADDRESS=WRFROM
043746  012777  000010  135270        MOV     #10,@RHDST      ;DESIRED TRACK=0 SECTOR=10
```

```
                                        BIS     #BAI,@RHCS2     ;BUS ADDRESS INCREMENT INHIBIT
                                        MOV     #FMT22,@RHOF    ;FORMAT 16 BIT WORDS
                                        CLR     @RHCA           ;CYLINDER=0
                                        MOV     #1,MID  ;SECTOR IS SET TO 1 SO THAT
                                                                ;WE CAN GET OUT AT THE
                                                                ;MIDDLE OF AN OPERATION
                                                                ;LOOKING FOR SECTOR 10
                                        MOV     #DMD,@RHMR      ;SET DIAGNOSTIC MODE
                                        BIS     #GO,@RHCS1              ;GO TO RHCS1 WITH 62
                                        JSR     R1,SEARCH
                                MID:    .WORD   0               ;SECTOR
                                        MOV     (SP)+,R1        ;;POP STACK INTO R1
                                        MOV     (SP)+,R0        ;;POP STACK INTO R0
                                        RTS     PC

                                .SBTTL  JAM CURRENT CYLINDER ROUTINE

                                ;THIS SUBROUTINE WILL CHANGE THE CURRENT CYLINDER REGISTER
                                ;THIS IS DONE BY GIVING A SEEK COMMAND THEN AN INIT
                                ;WHICH WILL LOAD THE CURRENT CYLINDER WITH THE DESIRED CYLINDER VALUE
                                ;
                                ;CALL
                                ;       JSR     R0,MAKECYL
                                ;       .WORD   XC              ;DESIRED VALUE OF CURRENT CYLINDER

                                MAKECYL:
                                        MOV     R5,-(SP)        ;;PUSH R5 ON STACK
                                        MOV     R0,PCJSR        ;PC OF JSR+4
                                        SUB     #4,PCJSR        ;SAVE PC OF JSR
                                        MOV     (R0)+,R5        ;GETTING READY TO FILL DESIRED CYLINDER
                                        MOV     R5,@RHCA        ;FILL DESIRED CYLINDER REGISTER
                                        CLR     @RHDST          ;MAKE SURE DESIRED SECTOR TRACK IS NOT ILLEGAL
                                        MOV     SEECOM,@RHCS1   ;FILL SEEK COMMAND
                                        MOV     #DMD,@RHMR      ;SET DIAGNOSTIC MODE
                                        BIS     #GO,@RHCS1      ;GO TO SEEK
                                        NOP                     ;ALLOW TIME FOR SEEK TO HANG UP
                                        NOP                     ;ALLOW TIME FOR SEEK TO HANG UP
                                        NOP                     ;ALLOW TIME FOR SEEK TO HANG UP
                                        NOP                     ;ALLOW TIME FOR SEEK TO HANG UP
                                        JSR     PC,CLDISK       ;GIVE INIT
                                        MOV     @RHCC,$BDDAT    ;TEST DATA
                                        CMP     R5,$BDDAT       ;COMPARE CURRENT CYLINDER
                                        BEQ     1$              ;BRANCH IF GOOD
                                        MOV     R5,$GDDAT       ;GOOD VALUE OF RHCC
                                        MOV     RHCC,REGADR     ;FAILING REGISTER ADDRESS
                                        EMT     30
                                                                ;REGISTER AFTER A SEEK AND AN INIT
                                1$:
                                        MOV     (SP)+,R5        ;;POP STACK INTO R5
                                        RTS     R0

                                .SBTTL  ECC GENERATION AND COMPARISON ROUTINE

                                ;THIS SUBROUTINE GENERATES AND TESTS ECC
                                ;CALL
                                ;       JSR     PC,ECTEST
```

```
                    100000                  PIE1    =100000
                    040000                  PIE2    =40000
                    020000                  PIE3    =20000
                    010000                  PIE4    =10000
                    004000                  PIE5    =4000
                    002000                  PIE6    =2000
                    001000                  PIE7    =1000
                    000400                  PIE8    =400
                    000200                  PIE9    =200
                    000100                  PIE10   =100
                    000040                  PIE11   =40
                    000020                  PIE12   =20
                    000010                  PIE13   =10
                    000004                  PIE14   =4
                    000002                  PIE15   =2
                    000001                  PIE16   =1
                    100000                  PIE17   =100000
                    040000                  PIE18   =40000
                    020000                  PIE19   =20000
                    010000                  PIE20   =10000
                    004000                  PIE21   =4000
                    002000                  PIE22   =2000
                    001000                  PIE23   =1000
                    000400                  PIE24   =400
                    000200                  PIE25   =200
                    000100                  PIE26   =100
                    000040                  PIE27   =40
                    000020                  PIE28   =20
                    000010                  PIE29   =10
                    000004                  PIE30   =4
                    000002                  PIE31   =2
                    000001                  PIE32   =1

044152  000000      ECDATA: 0               ;DATA BIT FOR ECC
                                            ;IF ALL ONES THEN CURRENT BIT IS A ONE
                                            ;IF ZERO THEN CURRENT BIT IS A ZERO

044154  000000      GECC1:  0               ;LOW ORDER ECC WORD TO BE GENERATED HERE
                                            ;=R1

044156  000000      GECC2:  0               ;HIGH ORDER ECC WORD TO BE GENERATED HERE
                                            ;=R2

044160  000000      TSECCG: 0               ;IF =177777 GENERATE AND TEST ECC FOR THIS BIT
                                            ;IF =0 DO NOT GENERATE AND TEST ECC FOR THIS BIT

044162  113713      NCODE:  38859.          ;N-CODE WORD
044164  000000      NCOUNT: 0               ;TEMPORARY N CODE
044166  000000      POSITI: 0               ;POSITION REGISTER
044170  010041      HARDER: 4129.           ;HARD ERROR COUNT
                                            ;TRUE COUNT IS 4128 BUT AS COMPARES ARE
                                            ;DONE ONE STAGE LATER SO 4129
044172  000000      DATENV: 0               ;DATA ENVELOPE FOR TYPE OUT
                                            ;MAX FOR WRITE IS 4096
                                            ;MAX FOR READ IS 4128
044174  000000      ZCODE:  0               ;LEADING ZEROS ENVELOPE FOR TYPE OUT
                                            ;THIS IS SHUT OFF WHEN POSITION COUNTER
```

```
                                                        ;IN ENABLED
                                                        ;MAX COUNT IS 38859


044176  000000                HADTMP: 0               ;TEMPORARY HARD ERROR COUNT
044200  000000                P3:     0
044202  000000                P12:    0
044204  000000                P22:    0
044206  000000                P24:    0


044210                      ECTEST:
044210  010046                      MOV     R0,-(SP)        ;;PUSH R0 ON STACK
044212  010146                      MOV     R1,-(SP)        ;;PUSH R1 ON STACK
044214  010246                      MOV     R2,-(SP)        ;;PUSH R2 ON STACK
044216  010346                      MOV     R3,-(SP)        ;;PUSH R3 ON STACK
044220  010446                      MOV     R4,-(SP)        ;;PUSH R4 ON STACK
044222  010546                      MOV     R5,-(SP)        ;;PUSH R5 ON STACK
044224  013701  044154              MOV     GECC1,R1                ;ECC1 WORD
044230  013702  044156              MOV     GECC2,R2                ;ECC2 WORD
044234  005737  044152              TST     ECDATA          ;IS CURRENT BIT A ONE
044240  001406                      BEQ     2$                      ;BRANCH IF CURRENT DATA D=0

                                    ;IF CARRY IS NOT ZERO THEN D=1
                                    ;INVERT X32 TO GIVE R0

044242  010103              1$:     MOV     R1,R3
044244  052703  177776              BIS     #^CPIE32,R3
044250  005103                      COM     R3
044252  010300                      MOV     R3,R0
044254  000404                      BR      3$

                                    ;IF CARRY IS ZERO THEN D=0
                                    ;X32 BECOMES R0
044256  010103              2$:     MOV     R1,R3
044260  042703  177776              BIC     #^CPIE32,R3
044264  010300                      MOV     R3,R0

044266  000241              3$:     CLC
044270  006000                      ROR     R0
044272  006000                      ROR     R0
044274  005700                      TST     R0
044276  001462                      BEQ     10$             ;BRANCH IF R0=0
                                    ;INVERT X2

044300  010203                      MOV     R2,R3
044302  052703  137777              BIS     #^CPIE2,R3
044306  005103                      COM     R3
044310  010337  044200              MOV     R3,P3
044314  006237  044200              ASR     P3

                                    ;INVERT X11
```

```
                                                ;INVERT X21

044340  010103                                          MOV     R1,R3
044342  052703  173777                                  BIS     #^CPIE21,R3
044346  005103                                          COM     R3
044350  010337  044204                                  MOV     R3,P22
044354  006237  044204                                  ASR     P22

                                                ;INVERT X23

044360  010103                                          MOV     R1,R3
044362  052703  176777                                  BIS     #^CPIE23,R3
044366  005103                                          COM     R3
044370  010337  044206                                  MOV     R3,P24
044374  006237  044206                                  ASR     P24

                                                ;NOW THAT R0 FOR POSITION 1
                                                ;         P3 FOR POSITION 3
                                                ;        P12 FOR POSITION 12
                                                ;        P22 FOR POSITION 22
                                                ;        P24 FOR POSITION 24
                                                ;ARE KNOWN THE ROTATE WILL BE DONE AND
                                                ;THESE BITS JAMED IN

044400  006002                                          ROR     R2
044402  006001                                          ROR     R1
044404  053700  044200                                  BIS     P3,R0
044410  053700  044202                                  BIS     P12,R0
044414  042702  120020                                  BIC     #PIE1!PIE3!PIE12,R2
044420  050002                                          BIS     R0,R2

044422  005000                                          CLR     R0
044424  053700  044204                                  BIS     P22,R0
044430  053700  044206                                  BIS     P24,R0
044434  042701  002400                                  BIC     #PIE22!PIE24,R1
044440  050001                                          BIS     R0,R1
044442  000404                                          BR      12$

                                                ;THE PROGRAM COMES HERE IF R0=0
                                                ;SO AFTER ROTATE R0 GETS PUT INTO POSITION 1

044444  006002                                10$:     ROR     R2
044446  006001                                         ROR     R1
044450  042702  100000                                 BIC     #PIE1,R2
044454  010137  044154                        12$:     MOV     R1,GECC1        ;SAVE ECC1
044460  010237  044156                                 MOV     R2,GECC2        ;SAVE ECC2
044464  005737  044160                                 TST     TSECCG  ;IS HARDWARE TO BE CHECKED
                                                                            ;IF =1777777 TEST HARDWARE
                                                                            ;IF = 0 DO NOT TEST HARDWARE
044470  001432                                         BEQ     14$             ;BRANCH IF HARDWARE NOT TO BE CHECKED
```

```
                                        ;CHECK HARDWARE
044472  032777  000400  134440            BIT     #SW8,@SWR       ;IS SWITCH 8 SET
044500  001005                            BNE     15$             ;BRANCH IF SW8 IS SET
044502  032777  000100  134430            BIT     #SW6,@SWR       ;IS SWITCH 6 SET
044510  001401                            BEQ     15$             ;BRANCH IF SW6 IS NOT SET
044512  000421                            BR      14$             ;IF SWITCH 8 IS NOT SET AND
                                                                ;SWITCH 6 IS SET THEN
                                                                ;DO NOT DO COMPARES
044514  010146                  15$:      MOV     R1,-(SP)        ;GOOD PATTERN REGISTER
044516  042716  174000                    BIC     #174000,(SP)    ;GET ONLY PATTERN BITS
044522  022677  134544                    CMP     (SP)+,@RHEC2    ;COMPARE PATTERN REGISTER
044526  001404                            BEQ     13$             ;BRANCH IF GOOD
                                        ;TO SAVE TIME
044530  004737  041064                    JSR     PC,PUTREG       ;SAVE REGISTERS
044534  104035                            EMT     35
044536  000407                            BR      14$             ;BRANCH OUT
044540  023777  044166  134522  13$:      CMP     POSITI,@RHEC1   ;COMPARE POISTION REGISTER
044546  001403                            BEQ     14$             ;BRANCH IF GOOD
                                        ;TO SAVE TIME
044550  004737  041064                    JSR     PC,PUTREG       ;SAVE REGISTERS
044554  104035                            EMT     35

                                                                ;"DATA ENVLOP" GIVES NUMBER OF CLOCK
                                                                ;PULSES FROM BEGINING OF COMMAND
                                                                ;THAT IS THE CLOCKS IN THE R/W DATA FIELD ENVELOPE
                                                                ;
                                                                ;IN A WRITE THERE ARE 10000 OCTAL CLOCKS
                                                                ;IN A READ THERE ARE 10040 OCTAL CLOCKS
                                                                ;
                                                                ;
                                                                ;"N-CODE ZEROS" GIVE THE NUMBER OF CLOCKS
                                                                ; GIVEN FOR THE LEADING ZEROS FIELD
                                                                ;MAX COUNT IS 113713 OCTAL
                                                                ;
                                                                ;"GOOD POSITION" GIVES NUMBER OF CLOCKS
                                                                ;GIVEN AFTER LEADING ZEROS WHICH IS FOR THE DATA
                                                                ;FIELD
                                                                ;MAX COUNT IS 10040 OR 10041 OCTAL


044556                          14$:
044556  012605                            MOV     (SP)+,R5        ;;POP STACK INTO R5
044560  012604                            MOV     (SP)+,R4        ;;POP STACK INTO R4
044562  012603                            MOV     (SP)+,R3        ;;POP STACK INTO R3
044564  012602                            MOV     (SP)+,R2        ;;POP STACK INTO R2
044566  012601                            MOV     (SP)+,R1        ;;POP STACK INTO R1
044570  012600                            MOV     (SP)+,R0        ;;POP STACK INTO R0
044572  000207                            RTS     PC
```

```
                                ;THIS SUBROUTINE WILL CONTROL THE ECC GENERATION ROUTINE
                                ;FOR ERROR CORRECTION PROCESS
                                ;CALL
                                ;       JSR     PC,ECORR
                                ;       .WORD   XP              ;EXPECTED POSITION REGISTER WHEN
                                ;                               ;CORRECTION IS COMPLETE

044574  000000                  ERPOS:  0                       ;POSITION REG. WHEN CORRECTION IS COMPLETE

044576  010037  001414          ECORR:  MOV     R0,PCJSR        ;SAVE PC OF JSR + 4
044602  162737  000004  001414          SUB     #4,PCJSR        ;SAVE PC OF JSR
044610  012037  044574                  MOV     (R0)+,ERPOS     ;GET POSITION REG. WHEN CORRECTION IS COMPLETE
044614  010146                          MOV     R1,-(SP)        ;;PUSH R1 ON STACK
044616  013701  001260                  MOV     RHMR,R1 ;MAIMTENANCE REGISTER
044622  012711  000001                  MOV     #DMD,@R1        ;SET DIAGNOSTIC MODE BIT
044626  005037  044152                  CLR     ECDATA  ;ECC DATA IS ZERO


044632  005737  044166          1$:     TST     POSITI  ;IS  SOFTWARE POSITION NON ZERO
044636  001007                          BNE     2$              ;BRANCH IF N-CODE S COMPLETE
044640  005337  044164                  DEC     NCOUNT  ;DECREMENT N-CODE
044644  001001                          BNE     6$              ;BRANCH IF N-CODE IS NOT COMPLETE
044646  000403                          BR      2$              ;BRANCH  AS N-CODE IS COMPLETE
044650  005237  044174          6$:     INC     ZCODE           ;INCREMENT CLOCKS GIVEN FOR LEADING ZEROS
044654  000420                          BR      3$              ;BRANCH AS N-CODE IS NOT COMPLETE
                                                                ;GO TO GIVE CLOCK AND TEST ECC
044656  005237  044166          2$:     INC     POSITI  ;INCREMENT SOFTWARE POSITION
044662  023737  044574  044166          CMP     ERPOS,POSITI;HAVE ENOUGH CLOCKS BEEN GIVEN TO DETECT ERROR
044670  103012                          BHIS    3$              ;BRANCH IF MORE CLOCKS TO BE GIVEN
044672  023737  044176  044166          CMP     HADTMP,POSITI;HAVE ENOUGH CLOCKS BEEN GIVEN FOR HARD ERROR
                                                                ;THAT IS HAVE 4128 MORE CLOCKS BEEN GIVEN
044700  001415                          BEQ     5$              ;BRANCH IF YES
044702  032711  000400                  BIT     #ZER,@R1        ;CHECK ZERO DETECT BIT IN RHMR
044706  001016                          BNE     4$              ;BRANCH IS ZER SET
                                ;TO SAVE TIME
044710  004737  041064                  JSR     PC,PUTREG       ;SAVE REGISTERS
044714  104034                          EMT     34
                                                                ;WHEN 21 BITS IN ECC 32 BIT REGISTER IS  0


044716  052711  000002          3$:     BIS     #MCLK,@R1       ;SET CLOCK
044722  042711  000002                  BIC     #MCLK,@R1       ;CLEAR CLOCK
044726  004737  044210                  JSR     PC,ECTEST       ;GO TO GENERATE AND TEST ECC
044732  000737                          BR      1$              ;CONTINUE

                                ;THIS EXTRA CLOCK IS TO BRING ECH HIGH
                                ;AFTER THIS CLOCK POSITION REGISTER MAY BE 10040 OR 10041 OCTAL

044734  052711  000002          5$:     BIS     #MCLK,@R1       ;SET CLOCK
044740  042711  000002                  BIC     #MCLK,@R1       ;CLEAR CLOCK

044744                          4$:
044744  012601                          MOV     (SP)+,R1        ;;POP STACK INTO R1
044746  000200                          RTS     R0
```

```
                                ;THIS SUBROUTINE GENERATES THE ECC FOR WHAT IS ON DISK AND INSERTS THEM
                                ;ON LOCATIONS 'DISK+1000' AND 'DISK+1002'
                                
044750                          FILLEC:
044750  010046                          MOV     R0,-(SP)        ;;PUSH R0 ON STACK
044752  010146                          MOV     R1,-(SP)        ;;PUSH R1 ON STACK
044754  010246                          MOV     R2,-(SP)        ;;PUSH R2 ON STACK
044756  010346                          MOV     R3,-(SP)        ;;PUSH R3 ON STACK
044760  010446                          MOV     R4,-(SP)        ;;PUSH R4 ON STACK
044762  010546                          MOV     R5,-(SP)        ;;PUSH R5 ON STACK
044764  005037  044166                  CLR     POSITI  ;CLEAR POSITION
044770  005037  044154                  CLR     GECC1           ;CLEAR GECC1
044774  005037  044156                  CLR     GECC2           ;CLEAR
045000  012701  050224                  MOV     #DISK,R1        ;POINTER TO DATA FOR ECC GENERATION
045004  012702  000400                  MOV     #256.,R2        ;COUNTER FOR NUMBER OF DATA WORDS
045010  012703  000020          9$:     MOV     #16.,R3         ;COUNTER FOR NUMBER OF BITS PER WORD
045014  012104                  10$:    MOV     (R1)+,R4        ;DATA IN R4
045016  006004                          ROR     R4              ;GET ONE DATA BIT IN CARRY
045020  103004                          BCC     11$             ;BRANCH IF DATA BIT IS ZERO
045022  012737  177777  044152          MOV     #-1,ECDATA      ;ECC DATA BIT IS A ONE
045030  000402                          BR      12$             ;BRANCH TO GENERATE ECC
045032  005037  044152          11$:    CLR     ECDATA  ;ECC DATA BIT IS A ZERO
045036  004737  044210          12$:    JSR     PC,ECTEST       ;GO TO GENERATE ECC
045042  005303                          DEC     R3              ;DECREMENT BIT COUNT
045044  001364                          BNE     10$             ;BRANCH IF 16 BITS NOT DONE
045046  005302                          DEC     R2              ;DECREMENT WORD COUNT
045050  001357                          BNE     9$              ;BRANCH IF 256 WORDS NOT DONE
045052  013737  044154  051224          MOV     GECC1,DISK+<256.*2>;INSERT ECC1 ON DISK
045060  013737  044156  051226          MOV     GECC2,DISK+<257.*2>;INSERT ECC2 ON DISK
045066  012605                          MOV     (SP)+,R5        ;;POP STACK INTO R5
045070  012604                          MOV     (SP)+,R4        ;;POP STACK INTO R4
045072  012603                          MOV     (SP)+,R3        ;;POP STACK INTO R3
045074  012602                          MOV     (SP)+,R2        ;;POP STACK INTO R2
045076  012601                          MOV     (SP)+,R1        ;;POP STACK INTO R1
045100  012600                          MOV     (SP)+,R0        ;;POP STACK INTO R0
045102  000207                  RTS     PC
```

```
                                .SBTTL  RH BASE ADDRESS CHANGE ROUTINE

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

045104  012706  001100      BASECH: MOV     #STACK,SP       ;RESET STACK
045110  104401  045116              TYPE    ,65$            ;;TYPE ASCIZ STRING
045114  000415                      BR      64$             ;;GET OVER THE ASCIZ
                        ;;65$:  .ASCIZ  <CRLF>/PRESENT BASE ADDRESS IS /
045150                  64$:
045150  013746  001240              MOV     RHCS1,-(SP)     ;GET READY TO TYPE OLD BASE
045154  104402                      TYPOC
045156  104401  045164              TYPE    ,67$            ;;TYPE ASCIZ STRING
045162  000425                      BR      66$             ;;GET OVER THE ASCIZ
                        ;;67$:  .ASCIZ  <CRLF>/TYPE NEW BASE ADDRESS FOLLOWED BY 'CR' /
045236                  66$:
045236  004737  054216              JSR     PC,$TKINI       ;INITIALIZE THE TTY KEYBOARD
045242  104412                      RDOCT
045244  012700  001230              MOV     #RHDB,R0        ;GET STARTING ADDRESS OF REGISTERS
045250  012701  000026              MOV     #22.,R1         ;NUMBER OF REGISTERS
045254  012737  046034  000004      MOV     #ADTIMO,4       ;SET UP TO CHECK NEW ADDRESS
045262  021637  001240              CMP     (SP),RHCS1      ;NEW CSR?
045266  001407                      BEQ     1$              ;NO-SKIP NEXT
045270  005776  000000              TST     @(SP)           ;ACCESS THE NEW ADDRESS
045274  163716  001240              SUB     RHCS1,(SP)      ;GET THE ADDRESS OFFSET
045300  061620                  2$: ADD     (SP),(R0)+      ;AND PLUG IT IN
045302  005301                      DEC     R1              ;DONE ALL OF THEM YET?
045304  001375                      BNE     2$              ;NOT YET, SO DO MORE
045306                  1$:
045306  104401  045314              TYPE    ,69$            ;;TYPE ASCIZ STRING
045312  000416                      BR      68$             ;;GET OVER THE ASCIZ
                        ;;69$:  .ASCIZ  <CRLF>/PRESENT VECTOR ADDRESS IS /
045350                  68$:
045350  013746  001226              MOV     RPVEC,-(SP)     ;GET READY TO TYPE OLD VECTOR ADDRESS
045354  104402                      TYPOC
045356  104401  045364              TYPE    ,71$            ;;TYPE ASCIZ STRING
045362  000437                      BR      70$             ;;GET OVER THE ASCIZ
                        ;;71$:  .ASCIZ  <CRLF>/TYPE NEW VECTOR ADDRESS OR RETYPE OLD ONE FOLLOWED BY 'CR' /
045462                  70$:
045462  104412                      RDOCT
045464  012637  001226              MOV     (SP)+,RPVEC     ;SETUP VECTOR ADDRESS
045470  104401  045476              TYPE    ,73$            ;;TYPE ASCIZ STRING
045474  000413                      BR      72$             ;;GET OVER THE ASCIZ
                        ;;73$:  .ASCIZ  <CRLF>/NEW BASE ADDRESS IS /
045524                  72$:
045524  013746  001240              MOV     RHCS1,-(SP)
045530  104402                      TYPOC
045532  104401  045540              TYPE    ,75$            ;;TYPE ASCIZ STRING
045536  000414                      BR      74$             ;;GET OVER THE ASCIZ
                        ;;75$:  .ASCIZ  <CRLF>/NEW VECTOR ADDRESS IS /
045570                  74$:
045570  013746  001226              MOV     RPVEC,-(SP)
045574  104402                      TYPOC
045576  104401  045604              TYPE    ,77$            ;;TYPE ASCIZ STRING
045602  000416                      BR      76$             ;;GET OVER THE ASCIZ
                        ;;77$:  .ASCIZ  <CRLF>/UNTIL PROGRAM IS RELOADED./
045640                  76$:
```

```
RH BASE ADDRESS CHANGE ROUTINE

     36 045640  104401  045646                    TYPE    ,79$            ;;TYPE ASCIZ STRING
        045644  000402                            BR      78$             ;;GET OVER THE ASCIZ
                                           ;;79$:  .ASCIZ  <CRLF>/ /
        045652                             78$:
     37 045652  104401  045660                    TYPE    ,81$            ;;TYPE ASCIZ STRING
        045656  000424                            BR      80$             ;;GET OVER THE ASCIZ
                                           ;;81$:  .ASCIZ  <CRLF>/UNLESS HALTED AND MANUALLY RESTARTED,/
        045730                             80$:
     38 045730  104401  045736                    TYPE    ,83$            ;;TYPE ASCIZ STRING
        045734  000425                            BR      82$             ;;GET OVER THE ASCIZ
                                           ;;83$:  .ASCIZ  <CRLF>/PROGRAM WILL AUTOMATICALLY RESTART FROM /
        046010                             82$:
     39 046010  012746  000200                    MOV     #200,-(SP)
     40 046014  104402                            TYPOC
     41 046016  104401  046024                    TYPE    ,85$            ;;TYPE ASCIZ STRING
        046022  000402                            BR      84$             ;;GET OVER THE ASCIZ
                                           ;;85$:  .ASCIZ  <CRLF>/ /
        046030                             84$:
     42 046030  000137  004310                    JMP     BEGIN           ;OK, NOW START OVER WITH NEW ADDRESS
     43
     44 046034                             ADTIMO:
        046034  104401  046042                    TYPE    ,65$            ;;TYPE ASCIZ STRING
        046040  000423                            BR      64$             ;;GET OVER THE ASCIZ
                                           ;;65$:  .ASCIZ  <CRLF><377>/SPECIFIED ADDRESS DID NOT RESPOND. /
        046110                             64$:
     45 046110  022626                            CMP     (SP)+,(SP)+     ;RESTORE THE STACK
     46 046112  000137  045104                    JMP     BASECH          ;AND DO IT AGAIN.
     47
     48                                    ;THIS IS A LITTLE ROUTINE THAT TESTS NED BIT 11 IN RHCS2
     49                                    ;THIS LOOPS HERE FOR EVER
     50                                    ;TO BE USED ONLY IF DRIVES PRESENT LOOKING AT NED DOES NOT AGREE
     51                                    ;WITH WHAT IS REALY THERE
     52
     53 046116  000000                     ERUNIT: 0                       ;UNIT UNDER MANUAL TEST
     54
     55 046120  004737  041460             ERSTART:JSR     PC,CLDISK       ;SET GENERAL REG.
     56 046124  013712  046116                     MOV     ERUNIT,@R2      ;SELECT UNIT
     57 046130  005714                     1$:     TST     @R4             ;TEST RHER1
     58 046132  032712  010000                     BIT     #NED,@R2        ;TEST NED
     59 046136  001401                             BEQ     2$              ;BRANCH IF GOOD
     60 046140  000773                             BR      1$              ;NED NOT SET
     61 046142  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
                        ;CALL
                        ;       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
                        ;CALL
                        ;       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
                        ;CALL
                        ;       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
                        ;CALL
                        ;       JSR     PC,COMHD
                        ;
                        ;;*****************************************************************
                        ;WRITE DATA COMMAND
                        ;OR READ COMMAND I.E DATA ONLY OR HEADER AND DATA
                        ;;*****************************************************************
                        ;
                        ;THIS SUBROUTINE IS THE FIRST IN A SERIES OF NESTED SUBROUTINES
                        ;
                        ;IT ISSURE DIAGNOSTIC MODE, AN EXTRA DIAGNOSTIC INDEX, AND THE
                        ;'GO' BIT
                        ;
                        ;IT THEN CALL THREE OTHER SUBROUTINES, WHICH IN TURN CALL OTHER
```

```
                                        ;SUBROUTINES.  THESE ARE:
                                        ;
                                        ;SEARCH
                                        ;RDHEAD
                                        ;WRDATA
                                        ;REDATA
                                        ;
046144  000000                  RUNCTR: .WORD   0
                                        ;
046146  011637  001414          COMHD:  MOV     (SP),PCJSR      ;SAVE PC OF JSR + 4
046152  162737  000004  001414          SUB     #4,PCJSR        ;SAVE PC OF JSR
046160  010046                          MOV     R0,-(SP)        ;;PUSH R0 ON STACK
046162  010146                          MOV     R1,-(SP)        ;;PUSH R1 ON STACK
046164  010246                          MOV     R2,-(SP)        ;;PUSH R2 ON STACK
046166  010346                          MOV     R3,-(SP)        ;;PUSH R3 ON STACK
046170  010446                          MOV     R4,-(SP)        ;;PUSH R4 ON STACK
046172  010546                          MOV     R5,-(SP)        ;;PUSH R5 ON STACK
                                        ;
046174  012777  000001  133056          MOV     #DMD,@RHMR      ;SET DIAGNOSTIC MODE
046202  052777  000004  133050          BIS     #MINX,@RHMR     ;SET DIAGNOSTIC INDEX
046210  042777  000004  133042          BIC     #MINX,@RHMR     ;CLEAR DIAGNOSTIC INDEX
046216  052777  000001  133014          BIS     #GO,@RHCS1      ;ISSUE 'GO' BIT & STALL 'TILL 'RUN'
                                        ;(FUNCTION CODE IS ISSUED BY THE TEST)
046224  012737  000113  046144  RUNWAT: MOV     #75.,RUNCTR     ;LOAD STALL COUNT = APPROX. 450US
                                        ;FOR 11/50 CPU WITH CORE MEMORY
046232  005337  046144          1$:     DEC     RUNCTR          ;COUNT DOWN ONE
046236  001375                          BNE     1$              ;CONTINUE UNTIL = 0
                                        ;
046240  013746  046310                  MOV     SECOTR,-(SP)    ;GET DESIRED SECTOR/TRACK
046244  042716  177740                  BIC     #177740,(SP)    ;MAKE ONLY SECTOR
046250  012637  046260                  MOV     (SP)+,TRK       ;SAVE SECTOR
046254  004137  052354                  JSR     R1,SEARCH       ;ISSUE SECTOR CLOCKS
                                        ;
046260  000000                  TRK:    .WORD   0
046262  012701  000240                  MOV     #+NOP,R1        ;GOING TO MOVE NOPS
046266  010137  046320                  MOV     R1,SSYN ;NOP INTO SSYN
046272  010137  046322                  MOV     R1,HEDGAP       ;NOP INTO HEDGAP
046276  010137  046324                  MOV     R1,HEDSYN       ;NOP INTO HEDSYN
046302  004137  046430                  JSR     R1,RDHEAD
                                        ;
                                        ;DUMMY ERROR CALL LOCATIONS FOR THE READ HEADER OPERATION
                                        ;
046306  000000                  CYL:    .WORD   0               ;CYLINDER ADDRESS
046310  000000                  SECOTR: .WORD   0               ;SECTOR/TRACK ADDRESS
046312  000000                  KEY1:   .WORD   0               ;KEY1 WORD
046314  000000                  KEY2:   .WORD   0               ;KEY2 WORD
046316  000000                  X:      .WORD   0               ;X=1 WRITE COMMAND
                                                                ;X=0 READ COMMAND
                                        ;
046320  000240                  SSYN:   NOP                     ;IF "ERROR 2" INSERTED BY RDHEAD
                                                                ;SUBROUTINE THEN THE FIRST SYNC.
                                                                ;IS NOT DETECTED.  NO BAD DATA
                                                                ;IS GIVEN BECAUSE SYNC=144000
                                                                ;CANNOT BE READ.  WORD NO
                                                                ;IS "1" BECAUSE THIS IS THE FIRST
                                                                ;WORD TESTED
```

```
                                                                ;IF "ERROR 3" INSERTED BY
                                                                ;RDHEAD SUBROUTINE THEN THE
                                                                ;HEADER GAP 0'S WERE NOT
                                                                ;WRITTEN RIGHT.
                                                                ;IF "WORD NO" CONTAINS, SAY
                                                                ;3(8), THEN IT IS THE THIRD
                                                                ;WORD OF A 5 WORD HEADER
                                                                ;GAP THAT IS WRONG
                                                                ;"BAD DATA" CONTAINS WHAT IS
                                                                ;GOING ON THE DISK

046324  000240          HEDSYN: NOP                           ;IF "ERROR 3" INSERTED BY RDHEAD
                                                              ;SUBROUTINE THEN THE HEADER SYNC.
                                                              ;GENERATED BY DCL IS WRONG
                                                              ;OR THE LAST BYTE
                                                              ;OF THE HEADER GAP 0'S IS WRONG
                                                              ;IN EITHER CASE WORD NO=6
                                                              ;RIGHT BYTE IS HEADER 0
                                                              ;LEFT BYTE IS SYNC
                                                              ;"BAD DATA" HAS WHAT IS GOING
                                                              ;ON DISK

046326  005737  001406          TST     ERFLG$                ;ARE ANY ERRORS DETECTED
046332  001017                  BNE     OUT                   ;IF YES, EXIT
046334  005737  046316          TST     X                     ;IS IT A DATA WRITE ?
046340  001410                  BEQ     DAREAD                ;NO...THEN DO A DATA READ
046342  005737  046422          TST     NOSYNC                ;IS THIS FORCED HEADER ERROR COMMAND
                                                              ;IF YES  NOSYNC=-1 THEN WRITE OR READ
                                                              ;IS SHUT OFF SO BRANCH OUT
                                                              ;IF  NOSYNC=0 THEN CONTINUE
046346  001011                  BNE     OUT                   ;EXIT IF SET

046350  004137  047674          JSR     R1,WRDATA,            ;WRITE DATA

046354  000000          NOWORD: .WORD   0                     ;NO OF WORDS TO BE WRITTEN
046356  000000          Y:      .WORD   0                     ;
046360  000404                  BR      OUT                   ;EXIT

046362  004137  052630  DAREAD: JSR     R1,REDATA             ;READ DATA
046366  000000          DAWORD: .WORD   0                     ;NO OF WORDS TO BE READ
046370  000000                  .WORD   0
046372                  OUT:
046372  012605                  MOV     (SP)+,R5              ;;POP STACK INTO R5
046374  012604                  MOV     (SP)+,R4              ;;POP STACK INTO R4
046376  012603                  MOV     (SP)+,R3              ;;POP STACK INTO R3
046400  012602                  MOV     (SP)+,R2              ;;POP STACK INTO R2
046402  012601                  MOV     (SP)+,R1              ;;POP STACK INTO R1
046404  012600                  MOV     (SP)+,R0              ;;POP STACK INTO R0
046406  000207                  RTS     PC                    ;EXIT ROUTINE
```

```
                                ;;*****************************************************************
                                ;THE DISK SECTOR IS DEVIDED AS FOLLOWS
                                ;19 WORDS OF 0, ONE WORD 144000
                                ;THESE MAKE 39 BYTES FOR SECTOR GAP AND ONE SYNC. BYTE

046410  014400                  RSYNC:  14400
046412  000000                  RCYL:   0
046414  000000                  RSETR:  0
046416  000000                  RKEY1:  0
046420  000000                  RKEY2:  0

                                ;5 WORDS OF 0 ONE WORD 144000
                                ;THESE MAKE 11 BYTES FOR HEADER GAP AND ONE SYNC. BYTE
                                ;THESE ARE DCL GENERATED

                                ;THERE ARE 256 WORDS OF DATA
                                ;THERE ARE 2 WORDS FOR ECC GENERATED BY DCL
                                ;15 WORDS OF 0 FOR DATA GAP AND TOLERANCE GAP
                                ;;*****************************************************************

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

046422  000000                  NOSYNC: 0                       ;FORCED HEADER ERROR = -1
                                                                ;NORMAL = 0
046424  000000                  TY:     0                       ;ERROR TYPE NO.
046426  000000                  ERWORD: 0                       ;ERROR WORD NO.


046430  012137  046412  RDHEAD: MOV     (R1)+,RCYL              ;STORE CYLINDER ADDRESS
046434  012137  046414          MOV     (R1)+,RSETR             ;STORE SECTOR AND TRACK ADDRESS
046440  012137  046416          MOV     (R1)+,RKEY1             ;STORE KEY1
046444  012137  046420          MOV     (R1)+,RKEY2             ;STORE KEY2
046450  012137  047220          MOV     (R1)+,COMPA             ;STORE COMPARE OR NOT
046454  010146                  MOV     R1,-(SP)                ;;PUSH R1 ON STACK
046456  013700  001260          MOV     RHMR,R0                 ;R0 CONTAINS MAINTANENCE REG.
046462  012705  000002          MOV     #2,R5                   ;R5 IS A COUNTER FOR WORDS
046466  012710  000001          MOV     #DMD,@R0                ;DIAG. MODE
046472  052710  000010          BIS     #MSTCK,@R0              ;SET SECTOR FOR FIRST WORD
046476  052710  000002          BIS     #MCLK,@R0               ;SET CLOCK FOR FIRST WORD
046502  042710  000012          BIC     #MSTCK!MCLK,@R0         ;RESET SECTOR AND CLOCK
046506  000404                  BR      2$                      ;BRANCH OVER GIVING SECTOR FOR FIRST TIME
046510  012710  000013  1$:     MOV     #MSTCK!MCLK!DMD,@R0     ;SET SECTOR, CLOCK, DIAG. MODE, RESET INDEX
046514  042710  000012          BIC     #MSTCK!MCLK,@R0         ;RESET SECTOR, CLOCK
046520  012702  000007  2$:     MOV     #7,R2                   ;R2 IS A COUNTER FOR BYTES
046524  052710  000002  3$:     BIS     #MCLK,@R0               ;SET CLOCK
046530  042710  000002          BIC     #MCLK,@R0               ;RESET CLOCK
046534  005302                  DEC     R2                      ;BYTE COUNTER
046536  001372                  BNE     3$                      ;BRANCH IF BYTE NOT COMPLETE
046540  005305                  DEC     R5                      ;WORD COUNTER
046542  001362                  BNE     1$                      ;BRANCH IF WORD NOT COMPLETE
046544  012702  000022          MOV     #18.,R2                 ;NO OF WORDS OF ZEROS
046550  005037  047216  4$:     CLR     WORD                    ;READ 0
```

```
 58 046554  004737  047222                    JSR     PC,READ         ;GO TO READ
 59 046560  005302                            DEC     R2              ;COUNT
 60 046562  001372                            BNE     4$
 61 046564  013737  046410  047216            MOV     RSYNC,WORD      ;SYNC. WORD
 62 046572  004737  047222                    JSR     PC,READ
 63 046576  032710  001000                    BIT     #DTSY,@R0       ;SYNC. BYTE DETECTED?
 64 046602  001012                            BNE     5$              ;BRANCH IF SYNC DETECTED
 65 046604  012737  000001  046426            MOV     #1,ERWORD       ;ERROR WORD NO
 66 046612  013737  046410  001124            MOV     RSYNC,$GDDAT    ;SYNC WORD
 67 046620  012737  104002  046320            MOV     #104002,SSYN    ;INSERT "ERROR 2" IN SSYN
 68 046626  000571                            BR      13$             ;BRANCH OUT
 69 046630  013737  046412  047216  5$:       MOV     RCYL,WORD       ;SETUP CYLINDER
 70 046636  004737  047222                    JSR     PC,READ         ;READ
 71 046642  013737  046414  047216            MOV     RSETR,WORD      ;SETUP SECTOR/TRACK
 72 046650  004737  047222                    JSR     PC,READ         ;READ
 73 046654  013737  046416  047216            MOV     RKEY1,WORD      ;SETUP KEY1
 74 046662  004737  047222                    JSR     PC,READ         ;READ
 75 046666  013737  046420  047216            MOV     RKEY2,WORD      ;SETUP KEY2
 76 046674  004737  047222                    JSR     PC,READ         ;READ
 77 046700  013737  050206  047216            MOV     WCRC,WORD       ;SETUP CRC
 78 046706  004737  047222                    JSR     PC,READ         ;READ
 79 046712  005737  001426                    TST     TESDTE          ;IS THIS A DRIVE TIMING ERROR
 80 046716  001135                            BNE     13$             ;BRANCH OUT IF YES
 81 046720  005737  047220                    TST     COMPA           ;IS THIS A READ OR WRITE COMMAND
 82 046724  001472                            BEQ     11$
 83 046726  012705  050210                    MOV     #HEGAP,R5       ;POINTER FOR HEADER GAP
 84 046732  012702  000005                    MOV     #5,R2           ;NO OF WORDS OF ZEROS
 85 046736  012737  000006  046426  6$:       MOV     #6,ERWORD       ;ERROR WORD NO SET
 86 046744  004737  047454                    JSR     PC,WRITE        ;FOR HEADER GAP
 87 046750  005737  047452                    TST     WWORD           ;TEST WRITTEN WORD
 88 046754  001413                            BEQ     7$              ;BRANCH IF GOOD THAT IS 0
 89 046756  160237  046426                    SUB     R2,ERWORD       ;WORD NO IN ERROR
 90 046762  005037  001124                    CLR     $GDDAT          ;GOOD WORD SHOULD BE 0
 91 046766  013737  047452  001126            MOV     WWORD,$BDDAT    ;BAD DATA
 92 046774  012737  104003  046322            MOV     #104003,HEDGAP  ;"ERROR 2" GOES IN HEDGAP
 93 047002  000503                            BR      13$             ;BRANCH OUT
 94 047004  013725  047452          7$:       MOV     WWORD,(R5)+     ;SAVE HEADER GAP
 95 047010  005302                            DEC     R2
 96 047012  001351                            BNE     6$
 97 047014  004737  047454                    JSR     PC,WRITE        ;WRITE HEADER (DATA) GAP SYNC
 98 047020  023737  046410  047452            CMP     RSYNC,WWORD
 99 047026  001426                            BEQ     10$
100 047030  005737  046422                    TST     NOSYNC          ;IS THIS FORCED HEADER ERROR COMMAND
101                                                                   ;IF YES  NOSYNC=-1 THEN WRITE OR READ
102                                                                   ;IS SHUT OFF SO BRANCH OUT
103                                                                   ;IF NO NOSYNC=0 THEN CONTINUE
104 047034  001406                            BEQ     14$             ;BRANCH IF TRUE ERROR
105 047036  005737  047452                    TST     WWORD           ;
106 047042  001420                            BEQ     10$             ;BRANCH IF GOOD
107 047044  005037  001124                    CLR     $GDDAT          ;IT SHOULD BE ZERO
108 047050  000403                            BR      15$             ;BRANCH TO TYPE ERROR
109 047052  013737  046410  001124  14$:       MOV     RSYNC,$GDDAT    ;GOOD DATA
110 047060  013737  047452  001126  15$:       MOV     WWORD,$BDDAT    ;BAD DATA
111 047066  012737  000006  046426            MOV     #6,ERWORD
112 047074  012737  104003  046324            MOV     #104003,HEDSYN
113 047102  000443                            BR      13$             ;BRANCH OUT
114 047104  013725  047452          10$:      MOV     WWORD,(R5)+     ;SAVE DATA SYNC.
```

```
    115 047110  000440                        BR      13$
    116
    117                                 ;READ COMMAND START FROM HERE
    118
    119 047112  012702  000005          11$:  MOV     #5,R2
    120 047116  005037  047216          12$:  CLR     WORD
    121 047122  004737  047222                JSR     PC,READ         ;READ HEADER GAP
    122 047126  005302                        DEC     R2              ;IS 5 HEADER GAP ZEROS COMPLETE
    123 047130  001372                        BNE     12$             ;IF NOT BRANCH
    124 047132  013737  046410  047216        MOV     RSYNC,WORD      ;SYNC WORD
    125 047140  004737  047222                JSR     PC,READ         ;READ HEADER (DATA) SYNC)
    126 047144  005737  046422                TST     NOSYNC
    127 047150  001404                        BEQ     16$             ;IF NOT ERROR COMMAND BRANCH
    128 047152  032710  001000                BIT     #DTSY,@R0       ;SYNC. DETECTED
    129 047156  001415                        BEQ     13$             ;IF ZERO BRANCH OUT
    130 047160  000403                        BR      17$             ;IF NOT ZERO BRANCH TO ERROR
    131 047162  032710  001000          16$:  BIT     #DTSY,@R0       ;SYNC. DETECTED?
    132 047166  001011                        BNE     13$             ;BRANCH IF YES
    133 047170  012737  000006  046426  17$:  MOV     #6,ERWORD       ;ERROR WORD NO.
    134 047176  013737  046410  001124        MOV     RSYNC,$GDDAT;SYNC WORD
    135 047204  012737  104002  046324        MOV     #104002,HEDSYN
    136 047212                          13$:
        047212  012601                        MOV     (SP)+,R1        ;;POP STACK INTO R1
    137 047214  000201                        RTS     R1
```

```
                                ;;*****************************************************************
                                ;READ ONE WORD IN 'WORD''
                                ;;*****************************************************************

047216  000000                  WORD:   0
047220  000000                  COMPA:  0

047222                          READ:
047222  010246                          MOV     R2,-(SP)        ;;PUSH R2 ON STACK
047224  012705  000002                  MOV     #2,R5           ;WORD COUNTER
047230  012710  000001                  MOV     #DMD,@R0        ;SET DIAG. MODE
047234  006037  047216                  ROR     WORD            ;CHECKING IF THERE IS A ONE
047240  103002                          BCC     1$              ;IF NO ONE BRANCH
047242  052710  000020                  BIS     #MRD,@R0        ;SET BIT 4 IF DATA HAS ONE
047246  012702  000007          1$:     MOV     #7,R2           ;BYTE COUNTER
047252  052710  000012                  BIS     #MSTCK!MCLK,@R0 ;SET CLOCK,DATA IF ANY, SECTOR
047256  005737  044160                  TST     TSECCG          ;IS THIS BIT TO GENERATE AND TEST ECC
047262  001411                          BEQ     6$              ;BRANCH IF NO
047264  032710  000020                  BIT     #MRD,@R0        ;IS DATA BIT A ONE
047270  001404                          BEQ     5$              ;BRANCH IF DATA BIT IS 0
047272  012737  177777  044152          MOV     #-1,ECDATA      ;ECC DATA BIT IS A ONE
047300  000402                          BR      6$              ;BRANCH
047302  005037  044152          5$:     CLR     ECDATA          ;ECC DATA BIT IS A 0
047306  012746  000001          6$:     MOV     #DMD,-(SP)      ;KEEP ONLY DIAG. MODE
047312  006037  047216                  ROR     WORD            ;CHECKING IF THERE IS A ONE
047316  103002                          BCC     2$              ;IF NO ONE BRANCH
047320  012716  000021                  MOV     #MRD!DMD,(SP)   ;KEEP DATA AND DIAG. MODE
047324  012610                  2$:     MOV     (SP)+,@R0       ;PUT IN DATA,RESET CLOCK, SECTOR
047326  005737  044160                  TST     TSECCG          ;IS ECC TO BE GENERATED FOR THIS BIT
047332  001404                          BEQ     3$              ;BRANCH IF NO
047334  005237  044172                  INC     DATENV          ;NUMBER OF CLOCKS GIVEN FOR DATA ENVELOPE
047340  004737  044210                  JSR     PC,ECTEST       ;GO TO GENERATE AND TEST ECC
047344  052710  000002          3$:     BIS     #MCLK,@R0       ;SET CLOCK
047350  005737  044160                  TST     TSECCG          ;IS THIS BIT TO GENERATE ECC
047354  001411                          BEQ     8$              ;BRANCH IF NO
047356  032710  000020                  BIT     #MRD,@R0        ;IS DATA BIT A ONE
047362  001404                          BEQ     7$              ;BRANCH IF DATA BIT IS = 0
047364  012737  177777  044152          MOV     #-1,ECDATA      ;ECC DATA BIT IS A ONE
047372  000402                          BR      8$              ;BRANCH
047374  005037  044152          7$:     CLR     ECDATA          ;ECC DATA BIT IS = 0
047400  012746  000001          8$:     MOV     #DMD,-(SP)      ;KEEP DIAG. MODE
047404  006037  047216                  ROR     WORD            ;CHECKING IF THERE IS A ONE
047410  103002                          BCC     4$              ;BRANCH IF NO ONE
047412  012716  000021                  MOV     #MRD!DMD,(SP)   ;KEEP DIAG. MODE AND DATA
047416  012610                  4$:     MOV     (SP)+,@R0       ;SET DATA, DIAG. MODE, CLEAR CLOCK
047420  005737  044160                  TST     TSECCG          ;IS THIS BIT TO GENERATE ECC
047424  001404                          BEQ     9$              ;BRANCH IF NO
047426  005237  044172                  INC     DATENV          ;NUMBER OF CLOCKS GIVEN FOR DATA ENVELOPE
047432  004737  044210                  JSR     PC,ECTEST       ;GO TO GENERATE AND TEST ECC
047436  005302                  9$:     DEC     R2              ;BYTE COUNTER
047440  001341                          BNE     3$              ;BRANCH IF ONE BYTE NOT COMPLETE
047442  005305                          DEC     R5              ;WORD COUNTER
047444  001300                          BNE     1$              ;BRANCH IF ONE WORD NOT COMPLETE
047446  012602                          MOV     (SP)+,R2        ;;POP STACK INTO R2
047450  000207                          RTS     PC
```

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

047452  000000         WWORD:  0

047454                  WRITE:
047454  010046                  MOV     R0,-(SP)        ;;PUSH R0 ON STACK
047456  010246                  MOV     R2,-(SP)        ;;PUSH R2 ON STACK
047460  010346                  MOV     R3,-(SP)        ;;PUSH R3 ON STACK
047462  010546                  MOV     R5,-(SP)        ;;PUSH R5 ON STACK
047464  012705  000002          MOV     #2,R5           ;WORD COUNTER
047470  012710  000001          MOV     #1,@R0          ;SET DIAG. MODE
                                                        ;R0 HAS RHMR ADDRESS IN IT
047474  012702  000007  1$:     MOV     #7,R2           ;BYTE COUNTER
047500  012710  000013          MOV     #MSTCK!MCLK!DMD,@R0;SET SECTOR AND CLOCK
047504  032710  000040          BIT     #MWR,@R0        ;CHECK WRITE BIT IN MAINT. REG.
047510  001406                  BEQ     2$              ;BRANCH IF ZERO
047512  012737  177777  044152  MOV     #-1,ECDATA      ;ECC DATA BIT IS A ONE
047520  000261                  SEC                     ;SET CARRY
047522  006003                  ROR     R3              ;MOVE 1 FORWARD
047524  000404                  BR      3$
047526  005037  044152  2$:     CLR     ECDATA  ;ECC DATA BIT IS = 0
047532  000241                  CLC                     ;CLEAR CARRY
047534  006003                  ROR     R3              ;MOVE 0 FOR WWORD
047536  012710  000001  3$:     MOV     #DMD,@R0        ;CLEAR SECTOR AND CLOCK
047542  005737  044160          TST     TSECCG  ;IS THIS BIT TO GENERATE ECC
047546  001404                  BEQ     4$              ;BRANCH IF NO
047550  005237  044172          INC     DATENV  ;NUMBER OF CLOCKS GIVEN FOR DATA ENVELOPE
047554  004737  044210          JSR     PC,ECTEST       ;GO TO GENERATE AND TEST ECC
047560  052710  000002  4$:     BIS     #MCLK,@R0       ;SET CLOCK IN RHMR
047564  032710  000040          BIT     #MWR,@R0        ;CHECK WRITE BIT IN RHMR
047570  001406                  BEQ     5$              ;BRANCH IF ZERO
047572  012737  177777  044152  MOV     #-1,ECDATA      ;ECC DATA BIT IS A ONE
047600  000261                  SEC                     ;SET CARRY
047602  006003                  ROR     R3              ;MOVE 1 FOR WWORD
047604  000404                  BR      6$
047606  005037  044152  5$:     CLR     ECDATA          ;ECC DATA BIT IS ZERO
047612  000241                  CLC                     ;CLEAR CARRY
047614  006003                  ROR     R3              ;MOVE 0 FOR WWORD
047616  012710  000001  6$:     MOV     #DMD,@R0        ;CLEAR CLOCK
047622  005737  044160          TST     TSECCG          ;IS THIS BIT TO GENERATE ECC
047626  001404                  BEQ     7$              ;BRANCH IF NO
047630  005237  044172          INC     DATENV          ;NUMBER OF CLOCKS GIVEN FOR DATA ENVELOPE
047634  004737  044210          JSR     PC,ECTEST       ;GO TO GENERATE AND TEST ECC
047640  005302          7$:     DEC     R2              ;COUNT FOR BYTE END
047642  001346                  BNE     4$              ;IF NOT BYTE END BRANCH
047644  005305                  DEC     R5              ;COUNT FOR WORD END
047646  001312                  BNE     1$              ;IF NOT WORD END BRANCH

047650  010337  047452          MOV     R3,WWORD        ;STORE THE WORD

047654  012605                  MOV     (SP)+,R5        ;;POP STACK INTO R5
047656  012603                  MOV     (SP)+,R3        ;;POP STACK INTO R3
047660  012602                  MOV     (SP)+,R2        ;;POP STACK INTO R2
047662  012600                  MOV     (SP)+,R0        ;;POP STACK INTO R0
```

DISK SIMULATION

```
     51 047664  000207                          RTS     PC
```

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

047666  000000                  COUNTD: 0
047670  000400                  FORMAT: 256.
047672  000000                  ZWORDS: 0
047674                          WRDATA:
047674  011137  047666                  MOV     (R1),COUNTD     ;STORE NO. OF WORDS TO BE WRITTEN
047700  012102                          MOV     (R1)+,R2        ;SAME IN R2
047702  012137  047220                  MOV     (R1)+,COMPA     ;COMPARE OR NOT

047706  010046                          MOV     R0,-(SP)        ;;PUSH R0 ON STACK
047710  010146                          MOV     R1,-(SP)        ;;PUSH R1 ON STACK
047712  010246                          MOV     R2,-(SP)        ;;PUSH R2 ON STACK
047714  010346                          MOV     R3,-(SP)        ;;PUSH R3 ON STACK
047716  010446                          MOV     R4,-(SP)        ;;PUSH R4 ON STACK

047720  012701  000016                  MOV     #14.,R1         ;NO. OF TOLERANCE GAP WORDS
047724  012703  051232                  MOV     #TOLGAP,R3      ;START OF TOLERANCE GAP TABLE
047730  012723  177777          1$:     MOV     #-1,(R3)+       ;MAKE IT 177777
047734  005301                          DEC     R1              ;IS 14 COMPLETED
047736  001374                          BNE     1$              ;IF NO BRANCH
047740  013700  001260                  MOV     RHMR,R0         ;R0 CONTAINS MAINTANENCE REG.
047744  013746  047670                  MOV     FORMAT,-(SP)
047750  163716  047666                  SUB     COUNTD,(SP)
047754  011637  047672                  MOV     (SP),ZWORDS     ;NO. OF ZERO WORDS TO BE WRITTEN
047760  012604                          MOV     (SP)+,R4
047762  005737  001424                  TST     TSECC           ;IS THIS AN ECC TEST ?
047766  001403                          BEQ     7$              ;BRANCH IF NO
047770  012737  177777  044160          MOV     #-1,TSECCG      ;THESE BITS ARE TO GENERATE ECC
047776  012703  050224          7$:     MOV     #DISK,R3        ;ADDRESS THE "DISK" AREA
050002  004737  047454          2$:     JSR     PC,WRITE        ;WRITE INTO "WWORD"
050006  013723  047452                  MOV     WWORD,(R3)+     ;STORE ON SIMULATED DISK
050012  005302                          DEC     R2              ;COUNT DOWN
050014  001372                          BNE     2$              ;CONTINUE IF ALL WORDS NOT WRITTEN
050016  005704                          TST     R4              ;ANY ZEROS TO BE WRITTEN ?
050020  001406                          BEQ     4$              ;BRANCH IF NONE TO BE WRITTEN
050022  004737  047454          3$:     JSR     PC,WRITE        ;WRITE ZEROS INTO "WWORD"
050026  013723  047452                  MOV     WWORD,(R3)+     ;STORE INTO "DISK"
050032  005304                          DEC     R4
050034  001372                          BNE     3$
050036  005037  044160          4$:     CLR     TSECCG  ;NO MORE ECC TO BE GENERATED
050042  012701  000002                  MOV     #2,R1
050046  004737  047454          5$:     JSR     PC,WRITE        ;WRITE ECC1 AND ECC2 ON SIMULATED DISK
050052  013723  047452                  MOV     WWORD,(R3)+     ;STORE ON WEEC1 AND WEEC2
050056  005301                          DEC     R1
050060  001372                          BNE     5$
050062  004737  047454                  JSR     PC,WRITE        ;WRITE DATA GAP INTO "WWORD"
050066  013723  047452                  MOV     WWORD,(R3)+     ;STORE INTO "DISK"
050072  012701  000016                  MOV     #14.,R1
050076  004737  047454          6$:     JSR     PC,WRITE        ;WRITE TOLERANCE GAP ZEROS
050102  013723  047452                  MOV     WWORD,(R3)+     ;STORE INTO "DISK"
050106  005301                          DEC     R1
050110  001372                          BNE     6$
```

```
     54
     55 050112  012604                    MOV     (SP)+,R4        ;;POP STACK INTO R4
        050114  012603                    MOV     (SP)+,R3        ;;POP STACK INTO R3
        050116  012602                    MOV     (SP)+,R2        ;;POP STACK INTO R2
        050120  012601                    MOV     (SP)+,R1        ;;POP STACK INTO R1
        050122  012600                    MOV     (SP)+,R0        ;;POP STACK INTO R0
     56
     57 050124  000201                    RTS     R1
```

```
                                ;;*****************************************************************
                                ;WRITE HEADER AND DATA
                                ;
                                ;
                                ;THIS IS THE SIMULATED DISK
                                ;ONLY ONE SECTOR OF SPACE IS ALLOWED
                                ;;*****************************************************************

050126                          SECGAP: .BLKW   19.             ;SECTOR GAP 38 BYTES OF 0
050174                          WSSYNC: .BLKW   1               ;SECTOR GAP 1 BYTE OF 0 ONE SYNC BYTE
050176                          HEADER: .BLKW   4               ;HEADER = CYL, SECTOR/TRACK, KEY1, KEY2
050206                          WCRC:   .BLKW   1               ;CRC
050210                          HEGAP:  .BLKW   5               ;HEADER GAP 10 BYTES OF 0
050222                          HDWSYN: .BLKW   1               ;HEADER GAP 1 BYTE OF 0 ONE SYNC. BYTE
050224                          SILOTB:                         ;USED IN SILO TEST AS SILO TABLE
050224                          DISK:   .BLKW   256.            ;DATA SPACE
051224                          WECC1:  .BLKW   1               ;ECC1
051226                          WECC2:  .BLKW   1               ;ECC2
051230                          DTAGAP: .BLKW   1               ;DATA GAP 2 BYTES OF 0
051232                          TOLGAP: .BLKW   14.             ;TOLERANCE GAP 28 BYTES OF 0

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

                                ;THIS SUBROUTINE IS THE FIRST IN A SERIES OF NESTED SUBROUTINES

                                ;IT ISSUES DIAGNOSTIC MODE, AN EXTRA DIAGNOSTIC INDEX, AND THE
                                ;'GO' BIT

                                ;IT THEN CALL THREE OTHER SUBROUTINES, WHICH IN TURN CALL OTHER
                                ;SUBROUTINES.  THESE ARE:

                                ;SEARCH
                                ;WRHEAD
                                ;WRDATA

051266  000000                  RNCTR1: .WORD   0               ;'RUN' LINE STALL COUNTER
051270  011637  001414          COMWHD: MOV     (SP),PCJSR      ;SAVE PC OF JSR + 4
051274  162737  000004  001414          SUB     #4,PCJSR        ;SAVE PC OF JSR
051302  010046                        MOV     R0,-(SP)        ;;PUSH R0 ON STACK
051304  010146                        MOV     R1,-(SP)        ;;PUSH R1 ON STACK
051306  010246                        MOV     R2,-(SP)        ;;PUSH R2 ON STACK
051310  010346                        MOV     R3,-(SP)        ;;PUSH R3 ON STACK
051312  010446                        MOV     R4,-(SP)        ;;PUSH R4 ON STACK
051314  010546                        MOV     R5,-(SP)        ;;PUSH R5 ON STACK

051316  012777  000001  127734        MOV     #DMD,@RHMR      ;SET DIAGNOSTIC MODE
051324  052777  000004  127726        BIS     #MINX,@RHMR     ;SET DIAGNOSTIC INDEX
051332  042777  000004  127720        BIC     #MINX,@RHMR     ;CLEAR IT
051340  052777  000001  127672        BIS     #GO,@RHCS1      ;SET 'GO' BIT & STALL 'TILL 'RUN'

051346  012737  000113  051266  RNWAT1: MOV     #75.,RNCTR1     ;LOAD STALL COUNTER = APPROX. 450US
                                                                ;FOR 11/50 CPU WITH CORE MEMORY
051354  005337  051266          1$:     DEC     RNCTR1  ;COUNT DOWN 1 TIME
051360  001_`5                          BNE     1$              ;CONTINUE UNTIL 0
```

```
                                                    ;GET DESIRED SECTOR/TRACK
051362  013746  051446          MOV     WSECTR,-(SP)
051366  042716  177740          BIC     #177740,(SP)    ;MAKE ONLY SECTOR
051372  012637  051402          MOV     (SP)+,WTRK      ;SAVE SECTOR

051376  004137  052354          JSR     R1,SEARCH       ;ISSUE SECTOR CLOCKS
051402  000000          WTRK:   .WORD   0               ;SECTOR NO.

051404  012701  000240          MOV     #+NOP,R1        ;GOING TO MOVE NOPS
051410  010137  051456          MOV     R1,SEGPER       ;NOP INTO SEGAP
051414  010137  051460          MOV     R1,FSYNER       ;NOP INTP FSYNER
051420  010137  051462          MOV     R1,ERHEAD       ;NOP INTO ERHEAD
051424  010137  051464          MOV     R1,ERCRC        ;NOP INTO ERCRC
051430  010137  051466          MOV     R1,ERHDGP       ;NOP INTO ERHDGAP
051434  010137  051470          MOV     R1,HDESYN       ;NOP INTO HDESYN

051440  004137  051540          JSR     R1,WRHEAD       ;WRITE THE HEADER
051444  000000          WCYL:   0                       ;CYLINDER
051446  000000          WSECTR: 0                       ;SECTOR AND TRACK
051450  000000          WKEY1:  0                       ;KEY1
051452  000000          WKEY2:  0                       ;KEY2
051454  000000          GCRC:   0                       ;GOOD CRC

                        ;DUMMY ERROR CALL LOCATIONS FOR THE WRITE HEADER OPERATION

051456  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

051460  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.

051462  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
                                                        ;IF "ERROR 6" INSERTED BY
051464  000240                ERCRC:  NOP                 ;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.
051466  000240                ERHDGP: 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
051470  000240                HDESYN: NOP                 ;IF "ERROR 6" INSERTED BY
                                                        ;WRHEAD SUBROUTINE THEN LAST
                                                        ;HEADER GAP BYTE OR HEADER
                                                        ;SYNC BYTE GOING ON DISK IS WRONG.

                                                        ;WORD NO = 5
                                                        ;BAD DATA IS WHAT IS GOING
                                                        ;ON DISK, RIGHT BYTE IS HEADER
                                                        ;GAP 0 BYTE, LEFT BYTE IS HEADER
                                                        ;GAP SYNC.
051472  005737  001406              TST     ERFLG$      ;ARE ANY ERRORS DETECTED
051476  001004                      BNE     FOUT        ;IF YES EXIT

051500  004137  047674              JSR     R1,WRDATA   ;WRITE THE DATA
051504  000000          FNWORD:     .WORD   0           ;FORMAT COMMAND NO. OF DATA
051506  000000                      .WORD   0

051510                  FOUT:
051510  012605                      MOV     (SP)+,R5    ;;POP STACK INTO R5
051512  012604                      MOV     (SP)+,R4    ;;POP STACK INTO R4
051514  012603                      MOV     (SP)+,R3    ;;POP STACK INTO R3
051516  012602                      MOV     (SP)+,R2    ;;POP STACK INTO R2
051520  012601                      MOV     (SP)+,R1    ;;POP STACK INTO R1
051522  012600                      MOV     (SP)+,R0    ;;POP STACK INTO R0

051524  000207                      RTS     PC
```

DISK SIMULATION

```
                                ;;*****************************************************************
                                ;WRITE HEADER
                                ;;*****************************************************************

                                ;R0 = MAINT.REG.
                                ;R1 = SIMULATED DISK
                                ;R2 = BYTE COUNT
                                ;R3 = WRITE WORD
                                ;R5 = WORD COUNT

051526  000000                  SCYL:   0
051530  000000                  SSECTR: 0
051532  000000                  SKEY1:  0
051534  000000                  SKEY2:  0
051536  000000                  SCRC:   0

051540  012137  051526          WRHEAD: MOV     (R1)+,SCYL
051544  012137  051530                  MOV     (R1)+,SSECTR
051550  012137  051532                  MOV     (R1)+,SKEY1
051554  012137  051534                  MOV     (R1)+,SKEY2
051560  012137  051536                  MOV     (R1)+,SCRC
051564  010146                          MOV     R1,-(SP)        ;;PUSH R1 ON STACK
051566  012701  050126                  MOV     #SECGAP,R1      ;SIMULATED DISK INDICATOR
051572  013700  001260                  MOV     RHMR,R0         ;R0 NOW HAS MAINT. REG. ADDR.
051576  012710  000001                  MOV     #DMD,@R0        ;SET DIAG. MODE IN RHMR
051602  012705  000002                  MOV     #2,R5           ;WORD COUNTER
051606  052710  000010                  BIS     #MSTCK,@R0      ;SET SECTOR FOR FIRST BYTE
051612  012710  000013  1$:             MOV     #MSTCK!MCLK!DMD,@R0     ;SET SECTOR, CLOCK, DIAG. MODE, RESET INDEX
051616  032710  000040                  BIT     #MWR,@R0        ;CHECK WRITE BIT IN MAINT. REG.
051622  001403                          BEQ     2$
051624  000261                          SEC                     ;SET CARRY
051626  006003                          ROR     R3              ;MOVE ONE FORWARD
051630  000402                          BR      3$              ;
051632  000241                  2$:     CLC                     ;CLEAR CARRY
051634  006003                          ROR     R3              ;MOVE ZERO FORWARD
051636  012710  000001          3$:     MOV     #DMD,@R0        ;CLEAR CLOCK, SECTOR
051642  012702  000007                  MOV     #7,R2           ;BYTE COUNTER
051646  052710  000002          4$:     BIS     #MCLK,@R0       ;SET CLOCK
051652  032710  000040                  BIT     #MWR,@R0        ;CHECK WRITE BIT IN MAINT.REG.
051656  001403                          BEQ     5$              ;BRANCH IF ZERO
051660  000261                          SEC                     ;SET CARRY
051662  006003                          ROR     R3              ;MOVE ONE FORWARD
051664  000402                          BR      6$
051666  000241                  5$:     CLC
051670  006003                          ROR     R3
051672  012710  000001          6$:     MOV     #DMD,@R0        ;SET DIAG. MODE AGAIN IN RHMR
051676  005302                          DEC     R2
051700  001362                          BNE     4$
051702  005305                          DEC     R5
051704  001342                          BNE     1$              ;CONTINUE

051706  010321                          MOV     R3,(R1)+
051710  005703                          TST     R3
051712  001414                          BEQ     7$
051714  012737  000001  046426          MOV     #1,ERWORD
051722  005037  001124                  CLR     $GDDAT
```

```
     58 051726  010337  001126                MOV     R3,$BDDAT
     59 051732  012737  104006  051456        MOV     #104006,SEGPER
     60 051740  000137  052346                JMP     17$             ;BRANCH OUT
     61
     62 051744  012702  000022          7$:   MOV     #18.,R2         ;COUNT NO. OF SECTOR GAP
     63 051750  012737  000024  046426  10$:   MOV     #20.,ERWORD     ;COUNT TO GIVE ERROR WORD
     64 051756  004737  047454                JSR     PC,WRITE        ;WRITE SECTOR GAP
     65 051762  013721  047452                MOV     WWORD,(R1)+     ;STORE SECTOR GAP WORD
     66 051766  001413                        BEQ     11$
     67 051770  160237  046426                SUB     R2,ERWORD       ;IF NOT GET ERROR WORD NO.
     68 051774  005037  001124                CLR     $GDDAT  ;GOOD WORD
     69 052000  013737  047452  001126        MOV     WWORD,$BDDAT;BAD WORD
     70 052006  012737  104006  051456        MOV     #104006,SEGPER;STORE "ERROR 6" IN SEGPER
     71 052014  000554                        BR      17$             ;BRANCH OUT
     72
     73 052016  005302                  11$:   DEC     R2              ;HAVE 18 WORDS OF ZEROS BEEN WRITTEN ?
     74 052020  001353                        BNE     10$             ;IF NOT DO SO
     75
     76                                 ;AT THIS POINT THE SECTOR FOUND FLOP SHOULD
     77                                 ;BE HIGH.   SO THAT THE HEADER SYNC BYTE CAN BE GIVEN
     78
     79                                 ;HOWEVER IN THE DRIVE TIMING ERROR TEST THE REST OF THE ROUTINE
     80                                 ;IS ABORTED - HEADER SYNC BYTE IS NOT GIVEN
     81
     82 052022  005737  001426                TST     TESDTE  ;IS THIS A DRIVE TIMING ERROR
     83 052026  001147                        BNE     17$             ;BRANCH OUT IF YES
     84 052030  004737  047454                JSR     PC,WRITE        ;WRITE ONE SECTOR GAP 0 BYTE
     85                                                         ;AND ONE SYNC. BYTE = 230
     86 052034  013711  047452                MOV     WWORD,(R1)      ;SAVE 0 BYTE AND SYNC BYTE
     87 052040  023721  046410                CMP     RSYNC,(R1)+     ;IF SYNC. BYTE RIGHT
     88 052044  001414                        BEQ     12$             ;IF YES BRANCH
     89 052046  012737  000024  046426        MOV     #20.,ERWORD     ;IF NOT GET READY FOR ERROR
     90 052054  013737  046410  001124        MOV     RSYNC,$GDDAT;GOOD WORD
     91 052062  014137  001126                MOV     -(R1),$BDDAT    ;BAD WORD
     92 052066  012737  104006  051460        MOV     #104006,FSYNER;INSERT "ERROR 6" IN FSYNER
     93 052074  000524                        BR      17$             ;BRANCH OUT
     94
     95 052076  012702  000004          12$:  MOV     #4,R2           ;FOUR HEADER WORDS
     96 052102  012703  051526                MOV     #SCYL,R3        ;POINTER FOR HEADER TABLE
     97 052106  012737  000005  046426  13$:   MOV     #5,ERWORD       ;ERROR WORD NO SET
     98 052114  004737  047454                JSR     PC,WRITE        ;WRITE 4 HEADER WORDS
     99 052120  013711  047452                MOV     WWORD,(R1)      ;STORE WRITTEN WORD
    100 052124  022321                        CMP     (R3)+,(R1)+     ;IS IT RIGHT?
    101 052126  001412                        BEQ     14$             ;IF GOOD CONTINUE
    102                                                        ;IF NOT GET READY FOR PRINT
    103 052130  160237  046426                SUB     R2,ERWORD       ;WORD NO
    104 052134  014337  001124                MOV     -(R3),$GDDAT    ;GOOD DATA
    105 052140  014137  001126                MOV     -(R1),$BDDAT    ;BAD DATA
    106 052144  012737  104006  051462        MOV     #104006,ERHEAD;INSERT "ERROR 6"
    107 052152  000475                        BR      17$             ;BRANCH OUT
    108
    109 052154  005302                  14$:  DEC     R2              ;ARE 4 HEADER WORDS DONE?
    110 052156  001353                        BNE     13$             ;IF NOT DO THEM
    111 052160  004737  047454                JSR     PC,WRITE        ;WRITE CRC
    112 052164  013711  047452                MOV     WWORD,(R1)      ;STORE CRC
    113 052170  022137  051454                CMP     (R1)+,GCRC      ;COMPARE GOOD CRC
    114 052174  001414                        BEQ     20$             ;BRANCH IF GOOD
```

```
115 052176  014137  001126                  MOV     -(R1),$BDDATA   ;BAD CRC WRITTEN
116 052202  013737  051454  001124          MOV     GCRC,$GDDAT     ;GOOD CRC
117 052210  012737  000005  046426          MOV     #5,ERWORD       ;ERROR WORD NO
118 052216  012737  104006  051464          MOV     #104006,ERCRC   ;INSERT ERROR 6
119 052224  000450                          BR      17$             ;EXIT
120
121 052226  012702  000005          20$:    MOV     #5,R2           ;NO OF HEADER GAP
122 052232  012737  000006  046426  15$:    MOV     #6,ERWORD       ;ERROR WORD NO SET
123 052240  004737  047454                  JSR     PC,WRITE        ;WRITE HEADER GAP
124 052244  013721  047452                  MOV     WWORD,(R1)+     ;STORE
125 052250  001412                          BEQ     16$             ;IF GOOD BRANCH
126 052252  160237  046426                  SUB     R2,ERWORD       ;ERROR WORD NO
127 052256  005037  001124                  CLR     $GDDAT  ;GOOD DATA
128 052262  014137  001126                  MOV     -(R1),$BDDAT    ;BAD DATA
129 052266  012737  104006  051466          MOV     #104006,ERHDGP;STORE "ERROR 6"
130 052274  000424                          BR      17$             ;BRANCH OUT
131
132 052276  005302          16$:    DEC     R2              ;ARE 5 HEADER GAP ZEROS  DONE
133 052300  001354                          BNE     15$             ;IF NOT BRANCH
134 052302  004737  047454                  JSR     PC,WRITE
135 052306  013711  047452                  MOV     WWORD,(R1)
136 052312  023721  046410                  CMP     RSYNC,(R1)+
137 052316  001413                          BEQ     17$             ;EXIT
138 052320  012737  000005  046426          MOV     #5,ERWORD
139 052326  014137  001126                  MOV     -(R1),$BDDAT
140 052332  013737  046410  001124          MOV     RSYNC,$GDDAT
141 052340  012737  104006  051470          MOV     #104006,HDESYN
142
143 052346                  17$:
    052346  012601                          MOV     (SP)+,R1        ;;POP STACK INTO R1
144
145 052350  000201                          RTS     R1
```

```
                                ;;*****************************************************************
                                ;SEARCH SECTOR
                                ;;*****************************************************************

                                ;       R0=RHMR ADDRESS
                                ;       R1=PASSED ARGUMENT (SECTOR SEARCHED FOR)
                                ;       R2=CLOCK COUNT (PER BYTE)
                                ;       R3=SECTOR COUNTER FROM R1
                                ;       R5=BYTES PER WORD COUNT
                                ;BEBORE INDEX IS GIVEN TWO SECTOR CLOCKS ARE GIVEN TO RESET
                                ;SECTOR PULSE IN CASE IT IS SET
                                ;AT BEGINNING OF EACH SECTOR ONE SECTOR CLOCK HAS TO RISE
                                ;BEFORE CLOCK THEN EVERY EIGHT CLOCKS ONE SECTOR CLOCK IS
                                ;IDENTICAL WITH CLOCK
                                ;NUMBERING THE SECTOR CLOCKS AS FOLLOWS
                                ;THE SECTOR CLOCK UNDER INDEX - 0
                                ;THE NEXT - 1
                                ;THE NEXT  - 2
                                ;ETC.
                                ;THEN THE LAST SECTOR CLOCK IN ONE SECTOR HAS NUMBER - 608
                                ;THE NEXT SECTOR THEN HAS 608 SECTOR CLOCKS
                                ;THE NEXT SECTOR THEN HAS ANOTHER 608 SECTOR CLOCKS
                                ;AND SO ON

052352  000000                  SECTR:  0                       ;SECTOR SEARCHED FOR

052354  012137  052352          SEARCH: MOV     (R1)+,SECTR     ;SAVE SECTOR SEARCHED FOR
052360  010046                          MOV     R0,-(SP)        ;;PUSH R0 ON STACK
052362  010146                          MOV     R1,-(SP)        ;;PUSH R1 ON STACK
052364  010246                          MOV     R2,-(SP)        ;;PUSH R2 ON STACK
052366  010346                          MOV     R3,-(SP)        ;;PUSH R3 ON STACK
052370  010446                          MOV     R4,-(SP)        ;;PUSH R4 ON STACK
052372  010546                          MOV     R5,-(SP)        ;;PUSH R5 ON STACK
052374  013700  001260                  MOV     RHMR,R0         ;NOW R0 HAS MAINTENANCE REG. ADR.
052400  013703  052352                  MOV     SECTR,R3        ;SECTOR COUNTER
052404  012710  000001                  MOV     #DMD,@R0        ;SET DIAGNOSTIC MODE
052410  052710  000010                  BIS     #MSTCK,@R0      ;SET SECTOR CLOCK
052414  042710  000010                  BIC     #MSTCK,@R0      ;CLEAR SECTOR CLOCK
052420  052710  000010                  BIS     #MSTCK,@R0      ;SET SECTOR CLOCK
052424  042710  000010                  BIC     #MSTCK,@R0      ;CLEAR SECTOR CLOCK
                                                                ;THE ABOVE TWO SECTOR CLOCKS ARE GIVEN FOR
                                                                ;RESETING SECTOR PULSE
                                                                ;IN CASE IT STARTS SET
052430  052710  000014                  BIS     #MINX!MSTCK,@R0 ;SET INDEX AND SECTOR CLOCK
052434  012710  000001                  MOV     #DMD,@R0        ;RESET INDEX AND SECTOR CLOCK
052440  005703                          TST     R3              ;IF SECTOR REQUIRED JUMP OUT
052442  001461                          BEQ     7$              ;BRANCH OF SECTOR ZERO REQUIRED

                                ;NOW THE 304 WORDS WILL START

                                ;FOR FIRST BYTE SECTOR CLOCK WILL GO HIGH THEN CLOCK WILL GO HIGH
                                ;BOTH WILL COME DOWN TOGETHER THEN SEVEN CLOCKS WILL BE GIVEN
                                ;FOR SECOND BYTE AND ALL OTHER BYTES TILL NEXT SECTOR SECTOR CLOCK
                                ;WILL BE IDENTICAL WITH ONE CLOCK

                                ;ONE WORD ONLY
```

```
                                                
052444  012702  000010          1$:     MOV     #8.,R2          ;BYTE COUNTER
052450  012705  000002                  MOV     #2,R5           ;BYTES PER WORD
052454  052710  000010                  BIS     #MSTCK,@R0      ;SET SECTOR CLOCK
052460  052710  000002                  BIS     #MCLK,@R0       ;SET CLOCK
052464  000402                          BR      3$              ;BRANCH TO CLEAR SECTOR AND CLOCK
052466  052710  000012          2$:     BIS     #MSTCK!MCLK,@R0 ;SET SECTOR AND CLOCK
052472  042710  000012          3$:     BIC     #MSTCK!MCLK,@R0 ;CLEAR SECTOR AND CLOCK
052476  052710  000002          8$:     BIS     #MCLK,@R0       ;SET CLOCK
052502  042710  000002                  BIC     #MCLK,@R0       ;CLEAR CLOCK
052506  005302                          DEC     R2              ;BYTE COUNTER
052510  001372                          BNE     8$              ;BRANCH IF BYTE NOT COMPLETE
052512  012702  000007                  MOV     #7,R2           ;SETUP FOR SECOND BYTE
052516  005305                          DEC     R5              ;IS WORD COMPLETE?
052520  001362                          BNE     2$              ;BRANCH IF NOT COMPLETE
                                                                ;TO GIVE SECTOR CLOCK AND CLOCK

                                ;NOW 303 WORDS ARE LEFT AND ALL ARE IDENTICAL

052522  012701  000457                  MOV     #303.,R1        ;WORDS PER SECTOR COUNTER
052526  012705  000002          4$:     MOV     #2,R5           ;BYTES PER WORD COUNTER
052532  012702  000007          5$:     MOV     #7,R2           ;BYTE COUNTER (CLOCK COUNTER)
052536  052710  000012                  BIS     #MSTCK!MCLK,@R0 ;SET SECTOR CLOCK AND CLOCK
052542  042710  000012                  BIC     #MSTCK!MCLK,@R0 ;CLEAR SECTOR CLOCK AND CLOCK
052546  052710  000002          6$:     BIS     #MCLK,@R0       ;SET CLOCK
052552  042710  000002                  BIC     #MCLK,@R0       ;RESET CLOCK
052556  005302                          DEC     R2              ;IS BYTE COMPLETE?
052560  001372                          BNE     6$              ;BRANCH IF NOT COMPLETE
052562  005305                          DEC     R5              ;IS WORD COMPLETE?
052564  001362                          BNE     5$              ;BRANCH IF NOT
052566  005301                          DEC     R1              ;IS SECTOR COMPLETE
052570  001356                          BNE     4$              ;BRANCH IF NOT
052572  052710  000010                  BIS     #MSTCK,@R0      ;SET SECTOR
052576  042710  000010                  BIC     #MSTCK,@R0      ;CLEAR SECTOR
052602  005303                          DEC     R3              ;IS REQUIRED NO OF SECTORS COMPLETE
052604  001317                          BNE     1$              ;BRANCH IF NOT

052606                          7$:
052606  012605                          MOV     (SP)+,R5        ;;POP STACK INTO R5
052610  012604                          MOV     (SP)+,R4        ;;POP STACK INTO R4
052612  012603                          MOV     (SP)+,R3        ;;POP STACK INTO R3
052614  012602                          MOV     (SP)+,R2        ;;POP STACK INTO R2
052616  012601                          MOV     (SP)+,R1        ;;POP STACK INTO R1
052620  012600                          MOV     (SP)+,R0        ;;POP STACK INTO R0
052622  000201                          RTS     R1

                                ;;**************************************************************
                                ;READ ONE SECTOR OF DATA
                                ;;**************************************************************

052624  000000                  RNO:    0                       ;NO. OF WORDS READ
052626  000000                  RCOM:   0                       ;EXTRA STORAGE

052630  012137  052624          REDATA: MOV     (R1)+,RNO       ;SAVE NO. OF WORDS ONLY FOR INFORMATION
052634  012137  052626                  MOV     (R1)+,RCOM      ;EXTRA WORD ONLY FOR INFORMATION
052640  010146                          MOV     R1,-(SP)        ;;PUSH R1 ON STACK
052642  005737  001424                  TST     TSECC           ;IS THIS AN ECC TEST
```

```
                                                BEQ     1$              ;BRANCH IF NO
                                                MOV     #-1,TSECCG      ;THESE BITS ARE TO GENERATE ECC
                                        1$:     MOV     #258.,R2        ;256 WORDS PER SECTOR
                                                                        ;PLUS 2 ECC WORDS
                                                MOV     #DISK,R3        ;POINTE TO DISK SIMULATION
                                        2$:     MOV     (R3)+,WORD      ;READY TO READ CONTENTS
                                                JSR     PC,READ         ;READ
                                                DEC     R2              ;IS 256 WORDS DONE?
                                                BNE     2$              ;IF NOT BRANCH
                                                TST     TSECC           ;IS THIS AN ECC TEST
                                                BNE     4$              ;BRANCH OUT IF YES
                                                CLR     TSECCG          ;NO MORE ECC BITS ARE TO BE GENERATED
                                                MOV     #15.,R2         ;ONE DATA GAP, 14 TOLERANCE GAP
                                        3$:     MOV     (R3)+,WORD      ;READY TO READ CONTENTS OF WORD
                                                JSR     PC,READ ;READ
                                                DEC     R2              ;COUNT
                                                BNE     3$              ;BRANCH IF 14 NOT DONE
                                        4$:
                                                MOV     (SP)+,R1        ;;POP STACK INTO R1
                                                RTS     R1              ;RETURN

                                        RPVECT:
                                                TYPE    ,65$            ;;TYPE ASCIZ STRING
                                                BR      64$             ;;GET OVER THE ASCIZ
                                        ;;65$:  .ASCIZ  /UNEXPECTED RP INTERRUPT @ PC = /
                                        64$:
                                                TYPOC                           ;TYPE FROM PC
                                                MOV     #RPVECT,@RPVEC          ;RESTORE TRAP VECTOR
                                                RTI                             ;CHANGE TO CONTINUE
```

```
     1                                  .SBTTL  SCOPE HANDLER ROUTINE

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

     053020                             $SCOPE:
     053020  104407                             CKSWR                   ;;TEST  FOR CHANGE IN SOFT-SWR
     053022  005037  046422                     CLR     NOSYNC          ;CLEAR FLAG FOR HEADER ERROR COMMANDS
     053026  005037  001424                     CLR     TSECC           ;CLEAR FLAG FOR ECC TEST
                                                                        ;WHEN =177777 IT IS AN ECC TEST
                                                                        ;WHEN =0 IT IS NOT AN ECC TEST

     053032  005037  044160                     CLR     TSECCG          ;EVEN IN AN ECC TEST EVERY CLOCK
                                                                        ;IS NOT TO GENERATE ECC
                                                                        ;IF =177777 GENERATE ECC
                                                                        ;IF =0 DO NOT GENERATE ECC
     053036  005037  001426                     CLR     TESDTE          ;DRIVE TIMING ERROR TEST
     053042                             1$:
     053042  032777  040000  126070     1$:     BIT     #BIT14,@SWR     ;;LOOP ON PRESENT TEST?
     053050  001402                             BEQ     9$              ;;NO IF SW14=0
     053052  000137  053442                     JMP     $OVER           ;;JUMP OVER SCOPE ROUTINE
     053056                             9$:
                                        ;#####START OF CODE FOR THE XOR TESTER#####
     053056  000416                     $XTSTR: BR      6$              ;;IF RUNNING ON THE "XOR" TESTER CHANGE
                                                                        ;;THIS INSTRUCTION TO A "NOP" (NOP=240)
     053060  013746  000004                     MOV     @#ERRVEC,-(SP)  ;;SAVE THE CONTENTS OF THE ERROR VECTOR
     053064  012737  053104  000004             MOV     #5$,@#ERRVEC    ;;SET FOR TIMEOUT
     053072  005737  177060                     TST     @#177060        ;;TIME OUT ON XOR?
     053076  012637  000004                     MOV     (SP)+,@#ERRVEC  ;;RESTORE THE ERROR VECTOR
     053102  000544                             BR      $SVLAD          ;;GO TO THE NEXT TEST
     053104  022626                     5$:     CMP     (SP)+,(SP)+     ;;CLEAR THE STACK AFTER A TIME OUT
     053106  012637  000004                     MOV     (SP)+,@#ERRVEC  ;;RESTORE THE ERROR VECTOR
     053112  000504                             BR      7$              ;;LOOP ON THE PRESENT TEST
     053114                             6$:;#####END OF CODE FOR THE XOR TESTER#####
     053114  032777  000400  126016             BIT     #BIT08,@SWR     ;;LOOP ON SPEC. TEST?
     053122  001404                             BEQ     2$              ;;BR IF NO
     053124  127737  126010  001102             CMPB    @SWR,$TSTNM     ;;ON THE RIGHT TEST?    SWR<7:0>
     053132  001543                             BEQ     $OVER           ;;BR IF YES
     053134  105737  001103             2$:     TSTB    $ERFLG          ;;HAS AN ERROR OCCURRED?
     053140  001502                             BEQ     3$              ;;BR IF NO
     053142  022737  177777  056026             CMP     #-1,CPSAVE      ;;SEE IF TIMEOUT WAS PREVIOUSLY RECORDED
     053150  001455                             BEQ     2003$           ;;KICK AROUND ROUTINE IF SO
     053152  013746  000004                     MOV     ERRVEC,-(SP)    ;;SAVE CONTENTS OF ERROR VECTOR
     053156  012737  053174  000004             MOV     #2000$,ERRVEC   ;;SETUP 'TRAP' RETURN ADDRESS
     053164  013737  177766  056026             MOV     177766,CPSAVE   ;;MOVE CPU ERROR REGISTER TO CPSAVE FOR TEST
     053172  000406                             BR      2001$
     053174  012737  177777  056026     2000$:  MOV     #-1,CPSAVE      ;;SET CPU ERROR REGISTER TIMEOUT INDICATOR
     053202  012716  053210                     MOV     #2001$,(SP)     ;;SETUP RETURN ADDRESS
```

```
053206  000002                          RTI
053210  012637  000004          2001$:  MOV     (SP)+,ERRVEC    ;;RESTORE CONTENTS OF ERROR VECTOR

053214  022737  177777  056026  2002$:  CMP     #-1,CPSAVE      ;;SEE IF CPSAVE HAS CPU ERR REG TIMEOUT INDICATION
053222  001430                          BEQ     2003$           ;;BRANCH IF SO
053224  032737  000001  056026          BIT     #BIT00,CPSAVE   ;;SEE IF THE POWER MONITOR BIT IS ON
053232  001424                          BEQ     2003$           ;;BRANCH TO CONTINUE ROUTINE IF CLEAR
053234  042737  000001  177766          BIC     #BIT00,177766   ;;CLEAR THE BIT FOUND TO BE SET
053242  013746  001140                  MOV     SWR,-(SP)       ;;SAVE SWR ADDRESS
053246  017646  000000                  MOV     @(SP),-(SP)     ;;SAVE SWR VALUE
053252  012737  000176  001140          MOV     #176,SWR        ;;GET SOFTWARE SWR ADDRESS
053260  011677  125654                  MOV     (SP),@SWR       ;;GET CURRENT SWR VALUE
053264  042777  001000  125646          BIC     #BIT09,@SWR     ;;DON'T ALLOW LOOP ON ERROR ON THIS ERROR
053272  104177                          EMT     177             ;;CALL SPECIAL POWER FAIL BIT ERROR CALL
053274  012676  000000                  MOV     (SP)+,@(SP)     ;;RESTORE SWR TO ORIGINAL VALUE
053300  012637  001140                  MOV     (SP)+,SWR       ;;RESTORE SWR ADDRESS
053304                          2003$:
053304  123737  001115  001103          CMPB    $ERMAX,$ERFLG   ;;MAX. ERRORS FOR THIS TEST OCCURRED?
053312  101015                          BHI     3$              ;;BR IF NO
053314  032777  001000  125616          BIT     #BIT09,@SWR     ;;LOOP ON ERROR?
053322  001404                          BEQ     4$              ;;BR IF NO
053324  013737  001110  001106  7$:     MOV     $LPERR,$LPADR   ;;SET LOOP ADDRESS TO LAST SCOPE
053332  000443                          BR      $OVER
053334  105037  001103          4$:     CLRB    $ERFLG          ;;ZERO THE ERROR FLAG
053340  005037  001212                  CLR     $TIMES          ;;CLEAR THE NUMBER OF ITERATIONS TO MAKE
053344  000415                          BR      1$              ;;ESCAPE TO THE NEXT TEST
053346  032777  004000  125564  3$:     BIT     #BIT11,@SWR     ;;INHIBIT ITERATIONS?
053354  001011                          BNE     1$              ;;BR IF YES
053356  005737  001100                  TST     $PASS           ;;IF FIRST PASS OF PROGRAM
053362  001406                          BEQ     1$              ;;      INHIBIT ITERATIONS
053364  005237  001104                  INC     $ICNT           ;;INCREMENT ITERATION COUNT
053370  023737  001212  001104          CMP     $TIMES,$ICNT    ;;CHECK THE NUMBER OF ITERATIONS MADE
053376  002021                          BGE     $OVER           ;;BR IF MORE ITERATION REQUIRED
053400  012737  000001  001104  1$:     MOV     #1,$ICNT        ;;REINITIALIZE THE ITERATION COUNTER
053406  013737  053456  001212          MOV     $MXCNT,$TIMES   ;;SET NUMBER OF ITERATIONS TO DO
053414  105237  001102          $SVLAD: INCB    $TSTNM          ;;COUNT TEST NUMBERS
053420  011637  001106                  MOV     (SP),$LPADR     ;;SAVE SCOPE LOOP ADDRESS
053424  011637  001110                  MOV     (SP),$LPERR     ;;SAVE ERROR LOOP ADDRESS
053430  005037  001214                  CLR     $ESCAPE         ;;CLEAR THE ESCAPE FROM ERROR ADDRESS
053434  112737  000001  001115          MOVB    #1,$ERMAX       ;;ONLY ALLOW ONE(1) ERROR ON NEXT TEST
053442  013777  001102  125472  $OVER:  MOV     $TSTNM,@DISPLAY ;;DISPLAY TEST NUMBER
053450  013716  001106                  MOV     $LPADR,(SP)     ;;FUDGE RETURN ADDRESS
053454  000002                          RTI                     ;;FIXES PS
053456  000004          $MXCNT: 4                               ;;MAX. NUMBER OF ITERATIONS
```

```
                                .SBTTL  CONVERT BINARY TO DECIMAL AND TYPE ROUTINE

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

053460                          $TYPDS:
053460  010046                         MOV     R0,-(SP)        ;;PUSH R0 ON STACK
053462  010146                         MOV     R1,-(SP)        ;;PUSH R1 ON STACK
053464  010246                         MOV     R2,-(SP)        ;;PUSH R2 ON STACK
053466  010346                         MOV     R3,-(SP)        ;;PUSH R3 ON STACK
053470  010546                         MOV     R5,-(SP)        ;;PUSH R5 ON STACK
053472  012746  020200                 MOV     #20200,-(SP)    ;;SET BLANK SWITCH AND SIGN
053476  016605  000020                 MOV     20(SP),R5       ;;GET THE INPUT NUMBER
053502  100004                         BPL     1$              ;;BR IF INPUT IS POS.
053504  005405                         NEG     R5              ;;MAKE THE BINARY NUMBER POS.
053506  112766  000055  000001         MOVB    #'-,1(SP)       ;;MAKE THE ASCII NUMBER NEG.
053514  005000                1$:      CLR     R0              ;;ZERO THE CONSTANTS INDEX
053516  012703  053674                 MOV     #$DBLK,R3       ;;SETUP THE OUTPUT POINTER
053522  112723  000040                 MOVB    #' ,(R3)+       ;;SET THE FIRST CHARACTER TO A BLANK
053526  005002                2$:      CLR     R2              ;;CLEAR THE BCD NUMBER
053530  016001  053664                 MOV     $DTBL(R0),R1    ;;GET THE CONSTANT
053534  160105                3$:      SUB     R1,R5           ;;FORM THIS BCD DIGIT
053536  002402                         BLT     4$              ;;BR IF DONE
053540  005202                         INC     R2              ;;INCREASE THE BCD DIGIT BY 1
053542  000774                         BR      3$
053544  060105                4$:      ADD     R1,R5           ;;ADD BACK THE CONSTANT
053546  005702                         TST     R2              ;;CHECK IF BCD DIGIT=0
053550  001002                         BNE     5$              ;;FALL THROUGH IF 0
053552  105716                         TSTB    (SP)            ;;STILL DOING LEADING 0'S?
053554  100407                         BMI     7$              ;;BR IF YES
053556  106316                5$:      ASLB    (SP)            ;;MSD?
053560  103003                         BCC     6$              ;;BR IF NO
053562  116663  000001  177777         MOVB    1(SP),-1(R3)    ;;YES--SET THE SIGN
053570  052702  000060        6$:      BIS     #'0,R2          ;;MAKE THE BCD DIGIT ASCII
053574  052702  000040        7$:      BIS     #' ,R2          ;;MAKE IT A SPACE IF NOT ALREADY A DIGIT
053600  110223                         MOVB    R2,(R3)+        ;;PUT THIS CHARACTER IN THE OUTPUT BUFFER
053602  005720                         TST     (R0)+           ;;JUST INCREMENTING
053604  020027  000010                 CMP     R0,#10          ;;CHECK THE TABLE INDEX
053610  002746                         BLT     2$              ;;GO DO THE NEXT DIGIT
053612  003002                         BGT     8$              ;;GO TO EXIT
053614  010502                         MOV     R5,R2           ;;GET THE LSD
053616  000764                         BR      6$              ;;GO CHANGE TO ASCII
053620  105726                8$:      TSTB    (SP)+           ;;WAS THE LSD THE FIRST NON-ZERO?
053622  100003                         BPL     9$              ;;BR IF NO
053624  116663  177777  177776         MOVB    -1(SP),-2(R3)   ;;YES--SET THE SIGN FOR TYPING
053632  105013                9$:      CLRB    (R3)            ;;SET THE TERMINATOR
053634  012605                         MOV     (SP)+,R5        ;;POP STACK INTO R5
053636  012603                         MOV     (SP)+,R3        ;;POP STACK INTO R3
053640  012602                         MOV     (SP)+,R2        ;;POP STACK INTO R2
053642  012601                         MOV     (SP)+,R1        ;;POP STACK INTO R1
```

CONVERT BINARY TO DECIMAL AND TYPE ROUTINE

```
053644  012600                          MOV     (SP)+,R0        ;;POP STACK INTO R0
053646  104401  053674                  TYPE    ,$DBLK          ;;NOW TYPE THE NUMBER
053652  016666  000002  000004          MOV     2(SP),4(SP)     ;;ADJUST THE STACK
053660  012616                          MOV     (SP)+,(SP)
053662  000002                          RTI                     ;;RETURN TO USER
053664  023420                  $DTBL:  10000.
053666  001750                          1000.
053670  000144                          100.
053672  000012                          10.
053674                          $DBLK:  .BLKW   4
```

```
                                .SBTTL  TYPE ROUTINE

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

053704  105737  001157        $TYPE:  TSTB    $TPFLG          ;;IS THERE A TERMINAL?
053710  100002                        BPL     1$              ;;BR IF YES
053712  000000                        HALT                    ;;HALT HERE IF NO TERMINAL
053714  000407                        BR      3$              ;;LEAVE
053716  010046                1$:     MOV     R0,-(SP)        ;;SAVE R0
053720  017600  000002                MOV     @2(SP),R0       ;;GET ADDRESS OF ASCIZ STRING
053724  112046                2$:     MOVB    (R0)+,-(SP)     ;;PUSH CHARACTER TO BE TYPED ONTO STACK
053726  001005                        BNE     4$              ;;BR IF IT ISN'T THE TERMINATOR
053730  005726                        TST     (SP)+           ;;IF TERMINATOR POP IT OFF THE STACK
053732  012600                60$:    MOV     (SP)+,R0        ;;RESTORE R0
053734  062716  000002        3$:     ADD     #2,(SP)         ;;ADJUST RETURN PC
053740  000002                        RTI                     ;;RETURN
053742  122716  000011        4$:     CMPB    #HT,(SP)        ;;BRANCH IF <HT>
053746  001430                        BEQ     8$
053750  122716  000200                CMPB    #CRLF,(SP)      ;;BRANCH IF NOT <CRLF>
053754  001006                        BNE     5$
053756  005726                        TST     (SP)+           ;;POP  <CR><LF> EQUIV
053760  104401                        TYPE                    ;;TYPE A CR AND LF
053762  001223                        $CRLF
053764  105037  054172                CLRB    $CHARCNT        ;;CLEAR CHARACTER COUNT
053770  000755                        BR      2$              ;;GET NEXT CHARACTER
053772  004737  054054        5$:     JSR     PC,$TYPEC       ;;GO TYPE THIS CHARACTER
053776  123726  001156        6$:     CMPB    $FILLC,(SP)+    ;;IS IT TIME FOR FILLER CHARS.?
054002  001350                        BNE     2$              ;;IF NO GO GET NEXT CHAR.
054004  013746  001154                MOV     $NULL,-(SP)     ;;GET # OF FILLER CHARS. NEEDED
                                                              ;;AND THE NULL CHAR.
054010  105366  000001        7$:     DECB    1(SP)           ;;DOES A NULL NEED TO BE TYPED?
054014  002770                        BLT     6$              ;;BR IF NO--GO POP THE NULL OFF OF STACK
054016  004737  054054                JSR     PC,$TYPEC       ;;GO TYPE A NULL
054022  105337  054172                DECB    $CHARCNT        ;;DO NOT COUNT AS A COUNT
054026  000770                        BR      7$              ;;LOOP

                                ;HORIZONTAL TAB PROCESSOR

054030  112716  000040        8$:     MOVB    #' ,(SP)        ;;REPLACE TAB WITH SPACE
054034  004737  054054        9$:     JSR     PC,$TYPEC       ;;TYPE A SPACE
054040  132737  000007  054172        BITB    #7,$CHARCNT     ;;BRANCH IF NOT AT
054046  001372                        BNE     9$              ;;TAB STOP
054050  005726                        TST     (SP)+           ;;POP SPACE OFF STACK
054052  000724                        BR      2$              ;;GET NEXT CHARACTER
```

```
054054                          $TYPEC:
054054  105777  125064                  TSTB    @$TKS           ;;CHAR IN KYBD BUFFER?
054060  100022                          BPL     10$             ;;BR IF NOT
054062  017746  125060                  MOV     @$TKB,-(SP)     ;;GET CHAR
054066  042716  177600                  BIC     #177600,(SP)    ;;STRIP EXTRANEOUS BITS
054072  122716  000023                  CMPB    #$XOFF,(SP)     ;;WAS CHAR XOFF
054076  001012                          BNE     102$            ;;BR IF NOT
054100                          101$:
054100  105777  125040                  TSTB    @$TKS           ;;WAIT FOR CHAR
054104  100375                          BPL     101$
054106  117716  125034                  MOVB    @$TKB,(SP)      ;;GET CHAR
054112  042716  177600                  BIC     #177600,(SP)    ;;STRIP IT
054116  122716  000021                  CMPB    #$XON,(SP)      ;;WAS IT XON?
054122  001366                          BNE     101$            ;;BR IF NOT
054124                          102$:
054124  005726                          TST     (SP)+           ;;FIX STACK
054126                          10$:
054126  105777  125016                  TSTB    @$TPS           ;;WAIT UNTIL PRINTER IS READY
054132  100375                          BPL     10$
054134  116677  000002  125010          MOVB    2(SP),@$TPB     ;;LOAD CHAR TO BE TYPED INTO DATA REG.
054142  122766  000015  000002          CMPB    #CR,2(SP)       ;;IS CHARACTER A CARRIAGE RETURN?
054150  001003                          BNE     1$              ;;BRANCH IF NO
054152  105037  054172                  CLRB    $CHARCNT        ;;YES--CLEAR CHARACTER COUNT
054156  000406                          BR      $TYPEX          ;;EXIT
054160  122766  000012  000002  1$:     CMPB    #LF,2(SP)       ;;IS CHARACTER A LINE FEED?
054166  001402                          BEQ     $TYPEX          ;;BRANCH IF YES
054170  105227                          INCB    (PC)+           ;;COUNT THE CHARACTER
054172  000000          $CHARCNT:.WORD  0               ;;CHARACTER COUNT STORAGE
054174  000207          $TYPEX: RTS     PC
```

```
                                .SBTTL  TTY INPUT ROUTINE

                                ;;*********************************************************************
                                .ENABL  LSB
054176  000000                  $TKCNT: .WORD   0               ;;NUMBER OF ITEMS IN QUEUE
054200  000000                  $TKQIN: .WORD   0               ;;INPUT POINTER
054202  000000                  $TKQOUT: .WORD  0               ;;OUTPUT POINTER
054204                          $TKQSRT: .BLKB  9.              ;;TTY KEYBOARD QUEUE
        054215                  $TKQEND=.
                                .EVEN

                                ;*TK INITIALIZE ROUTINE
                                ;*THIS ROUTINE WILL INITIALIZE THE TTY KEYBOARD INPUT QUEUE
                                ;*SETUP THE INTERRUPT VECTOR AND TURN ON THE KEYBOARD INTERRUPT
                                ;
                                ;*CALL:
                                ;*      JSR     PC,$TKINT
                                ;*      RETURN
                                ;
054216  005037  054176          $TKINT: CLR     $TKCNT          ;;CLEAR COUNT OF ITEMS IN QUEUE
054222  012737  054204  054200          MOV     #$TKQSRT,$TKQIN ;;MOVE THE STARTING ADDRESS OF THE
054230  013737  054200  054202          MOV     $TKQIN,$TKQOUT  ;;QUEUE INTO THE INPUT & OUTPUT POINTERS.
054236  012737  054266  000060          MOV     #$TKSRV,@#TKVEC ;;INITIALIZE THE KEYBOARD VECTOR
054244  012737  000200  000062          MOV     #200,@#TKVEC+2  ;;'BR'' LEVEL 4
054252  005777  124670                  TST     @$TKB           ;;CLEAR DONE FLAG
054256  012777  000100  124660          MOV     #100,@$TKS      ;;ENABLE TTY KEYBOARD INTERRUPT
054264  000207                          RTS     PC              ;;RETURN TO CALLER

                                ;*TK SERVICE ROUTINE
                                ;*THIS ROUTINE WILL SERVICE THE TTY KEYBOARD INTERRUPT
                                ;*BY READING THE CHARACTER FROM THE INPUT BUFFER AND PUTTING
                                ;*IT IN THE QUEUE.
                                ;*IF THE CHARACTER IS A ''CONTROL-C'' (^C) $TKINT IS CALLED AND
                                ;*UPON RETURN EXIT IS MADE TO THE ''CONTROL-C'' RESTART ADDRESS (OPERSEL)
                                ;
054266  117746  124654          $TKSRV: MOVB    @$TKB,-(SP)     ;;PICKUP THE CHARACTER
054272  042716  177600                  BIC     #^C177,(SP)     ;;STRIP THE JUNK
054276  021627  000021                  CMP     (SP),#$XON      ;;IS IT A RANDOM XON?
054302  001002                          BNE     30$             ;;BRANCH IF NO
054304  005726                          TST     (SP)+           ;;CLEAN RANDOM XON OFF STACK
054306  000002                          RTI                     ;;RETURN
054310                          30$:
054310  021627  000003                  CMP     (SP),#3         ;;IS IT A CONTROL C?
054314  001007                          BNE     1$              ;;BRANCH IF NO
054316  104401  055267                  TYPE    ,$CNTLC         ;;TYPE A CONTROL-C (^C)
054322  004737  054216                  JSR     PC,$TKINT       ;;INIT THE KEYBOARD
054326  005726                          TST     (SP)+           ;;CLEAN UP STACK
054330  000137  040310                  JMP     OPERSEL         ;;CONTROL C RESTART
054334  021627  000007          1$:     CMP     (SP),#7         ;;IS IT A CONTROL G?
054340  001004                          BNE     2$              ;;BRANCH IF NO
054342  022737  000176  001140          CMP     #SWREG,SWR      ;;IS SOFT-SWR SELECTED?
054350  001500                          BEQ     6$              ;;GO TO SWR CHANGE

054352                          2$:
054352  022737  000011  054176          CMP     #9.,$TKCNT      ;;IS THE QUEUE FULL?
054360  001004                          BNE     3$              ;;BRANCH IF NO
054362  104401  001216                  TYPE    ,$BELL          ;;RING THE TTY BELL
```

```
054366  005726                          TST     (SP)+           ;;CLEAN CHARACTER OFF OF STACK
054370  000451                          BR      5$              ;;EXIT
054372  021627  000023          3$:     CMP     (SP),#23        ;;IS IT A CONTROL-S?
054376  001021                          BNE     32$             ;;BRANCH IF NO
054400  005077  124540                  CLR     @$TKS           ;;DISABLE TTY KEYBOARD INTERRUPTS
054404  005726                          TST     (SP)+           ;;CLEAN CHAR OFF STACK
054406  105777  124532          31$:    TSTB    @$TKS           ;;WAIT FOR A CHAR
054412  100375                          BPL     31$             ;;LOOP UNTIL ITS THERE
054414  117746  124526                  MOVB    @$TKB,-(SP)     ;;GET THE CHARACTER
054420  042716  177600                  BIC     #^C177,(SP)     ;;MAKE IT 7-BIT ASCII
054424  022627  000021                  CMP     (SP)+,#21       ;;IS IT A CONTROL-Q?
054430  001366                          BNE     31$             ;;BRANCH IF NO
054432  012777  000100  124504          MOV     #100,@$TKS      ;;REENABLE TTY KEYBOARD INTERRUPTS
054440  000002                          RTI                     ;;RETURN
054442  005237  054176          32$:    INC     $TKCNT          ;;COUNT THIS CHARACTER
054446  021627  000140                  CMP     (SP),#140       ;;IS IT UPPER CASE?
054452  002405                          BLT     4$              ;;BRANCH IF YES
054454  021627  000175                  CMP     (SP),#175       ;;IS IT A SPECIAL CHAR?
054460  003002                          BGT     4$              ;;BRANCH IF YES
054462  042716  000040                  BIC     #40,(SP)        ;;MAKE IT UPPER CASE
054466  112677  177506          4$:     MOVB    (SP)+,@$TKQIN   ;;AND PUT IT IN QUEUE
054472  005237  054200                  INC     $TKQIN          ;;UPDATE THE POINTER
054476  023727  054200  054215          CMP     $TKQIN,#$TKQEND ;;GO OFF THE END?
054504  001003                          BNE     5$              ;;BRANCH IF NO
054506  012737  054204  054200          MOV     #$TKQSRT,$TKQIN ;;RESET THE POINTER
054514  000002                  5$:     RTI                     ;;RETURN

                        ;;**************************************************************************
                        ;*SOFTWARE SWITCH REGISTER CHANGE ROUTINE.
                        ;*ROUTINE IS ENTERED  FROM THE TRAP HANDLER, AND WILL
                        ;*SERVICE THE TEST FOR CHANGE IN SOFTWARE SWITCH REGISTER TRAP
                        ;*CALL WHEN OPERATING IN TTY INTERRUPT MODE.
054516  022737  000176  001140  $CKSWR: CMP     #SWREG,SWR      ;;IS THE SOFT-SWR SELECTED
054524  001124                  BNE     15$             ;;EXIT IF NOT
054526  105777  124412          TSTB    @$TKS           ;;IS A CHAR WAITING?
054532  100121                  BPL     15$             ;;IF NOT, EXIT
054534  117746  124406          MOVB    @$TKB,-(SP)     ;;YES
054540  042716  177600          BIC     #^C177,(SP)     ;;MAKE IT  7-BIT ASCII
054544  021627  000007          CMP     (SP),#7         ;;IS IT A CONTROL-G?
054550  001300                  BNE     2$              ;;IF NOT, PUT IT IN THE TTY QUEUE
                                                        ;;AND EXIT

                        ;;**************************************************************************
                        ;*CONTROL IS PASSED TO THIS POINT FROM EITHER THE TTY INTERRUPT SERVICE
                        ;*ROUTINE OR FROM THE SOFTWARE SWITCH REGISTER TRAP CALL, AS A RESULT OF A
                        ;*CONTROL-G BEING TYPED, AND THE SOFTWARE SWITCH REGISTER BEING SELECTED.
054552  123727  001134  000001  6$:     CMPB    $AUTOB,#1       ;;ARE WE RUNNING IN AUTO-MODE?
054560  001674                  BEQ     2$              ;;BRANCH IF YES
054562  005726                  TST     (SP)+           ;;CLEAR CONTROL-G OFF STACK
054564  004737  054216          JSR     PC,$TKINT       ;;FLUSH THE TTY INPUT QUEUE
054570  005077  124350          CLR     @$TKS           ;;DISABLE TTY KEYBOARD INTERRUPTS
054574  112737  000001  001135  MOVB    #1,$INTAG       ;;SET INTERRUPT MODE INDICATOR

054602  104401  055301          TYPE    ,$CNTLG         ;;ECHO THE CONTROL-G (^G)
054606  104401  055306  $GTSWR: TYPE    ,$MSWR          ;;TYPE CURRENT CONTENTS
054612  013746  000176          MOV     SWREG,-(SP)     ;;SAVE SWREG FOR TYPEOUT
054616  104402                  TYPOC                   ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
```

```
054620  104401  055317          TYPE    ,$MNEW          ;;PROMPT FOR NEW SWR
054624  005046          19$:    CLR     -(SP)           ;;CLEAR COUNTER
054626  005046                  CLR     -(SP)           ;;THE NEW SWR
054630  105777  124310        7$:     TSTB    @$TKS           ;;CHAR THERE?
054634  100375                  BPL     7$              ;;IF NOT TRY AGAIN

054636  117746  124304          MOVB    @$TKB,-(SP)     ;;PICK UP CHAR
054642  042716  177600          BIC     #^C177,(SP)     ;;MAKE IT 7-BIT ASCII

054646  021627  000003          CMP     (SP),#3         ;;IS IT A CONTROL-C?
054652  001015                  BNE     9$              ;;BRANCH IF NOT
054654  104401  055267          TYPE    ,$CNTLC         ;;YES, ECHO CONTROL-C (^C)
054660  062706  000006          ADD     #6,SP           ;;CLEAN UP STACK
054664  123727  001135  000001  CMPB    $INTAG,#1       ;;REENABLE TTY KEYBOARD INTERRUPTS?
054672  001003                  BNE     8$              ;;BRANCH IF NO
054674  012777  000100  124242  MOV     #100,@$TKS      ;;ALLOW TTY KEYBOARD INTERRUPTS
054702  000137  040310      8$: JMP     OPERSEL         ;;CONTROL-C RESTART


054706  021627  000025      9$: CMP     (SP),#25        ;;IS IT A CONTROL-U?
054712  001005                  BNE     10$             ;;BRANCH IF NOT
054714  104401  055274          TYPE    ,$CNTLU         ;;YES, ECHO CONTROL-U (^U)
054720  062706  000006      20$:ADD     #6,SP           ;;IGNORE PREVIOUS INPUT
054724  000737                  BR      19$             ;;LET'S TRY IT AGAIN


054726  021627  000015      10$:CMP     (SP),#15        ;;IS IT A <CR>?
054732  001022                  BNE     16$             ;;BRANCH IF NO
054734  005766  000004          TST     4(SP)           ;;YES, IS IT THE FIRST CHAR?
054740  001403                  BEQ     11$             ;;BRANCH IF YES
054742  016677  000002  124170  MOV     2(SP),@SWR      ;;SAVE NEW SWR
054750  062706  000006      11$:ADD     #6,SP           ;;CLEAR UP STACK
054754  104401  001223      14$:TYPE    ,$CRLF          ;;ECHO <CR> AND <LF>
054760  123727  001135  000001  CMPB    $INTAG,#1       ;;RE-ENABLE TTY KBD INTERRUPTS?
054766  001003                  BNE     15$             ;;BRANCH IF NOT
054770  012777  000100  124146  MOV     #100,@$TKS      ;;RE-ENABLE TTY KBD INTERRUPTS
054776  000002      15$:RTI             ;;RETURN
055000  004737  054054      16$:JSR     PC,$TYPEC       ;;ECHO CHAR
055004  021627  000060          CMP     (SP),#60        ;;CHAR < 0?
055010  002420                  BLT     18$             ;;BRANCH IF YES
055012  021627  000067          CMP     (SP),#67        ;;CHAR > 7?
055016  003015                  BGT     18$             ;;BRANCH IF YES
055020  042726  000060          BIC     #60,(SP)+       ;;STRIP-OFF ASCII
055024  005766  000002          TST     2(SP)           ;;IS THIS THE FIRST CHAR
055030  001403                  BEQ     17$             ;;BRANCH IF YES
055032  006316                  ASL     (SP)            ;;NO, SHIFT PRESENT
055034  006316                  ASL     (SP)            ;;    CHAR OVER TO MAKE
055036  006316                  ASL     (SP)            ;;    ROOM FOR NEW ONE.
055040  005266  000002      17$:INC     2(SP)           ;;KEEP COUNT OF CHAR
055044  056616  177776          BIS     -2(SP),(SP)     ;;SET IN NEW CHAR
055050  000667                  BR      7$              ;;GET THE NEXT ONE
055052  104401  001222      18$:TYPE    ,$QUES          ;;TYPE ?<CR><LF>
055056  000720                  BR      20$             ;;SIMULATE CONTROL-U
                        .DSABL  LSB

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

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

055060  011646                  $RDCHR: MOV     (SP),-(SP)      ;;PUSH DOWN THE PC AND
055062  016666  000004  000002          MOV     4(SP),2(SP)     ;;THE PS
055070  005066  000004                  CLR     4(SP)           ;;GET READY FOR A CHARACTER
055074  005046                          CLR     -(SP)           ;;PUT NEW PS ON STACK
055076  012746  055104                  MOV     #64$,-(SP)      ;;PUT NEW PC ON  STACK
055102  000002                          RTI                     ;;POP NEW PC AND PS
055104                          64$:
055104  005737  054176          1$:     TST     $TKCNT          ;;WAIT ON A CHARACTER
055110  001775                          BEQ     1$
055112  005337  054176                  DEC     $TKCNT          ;;DECREMENT THE COUNTER
055116  117766  177060  000004          MOVB    @$TKQOUT,4(SP)  ;;GET ONE CHARACTER
055124  005237  054202                  INC     $TKQOUT         ;;UPDATE THE POINTER
055130  023727  054202  054215          CMP     $TKQOUT,#$TKQEND ;;DID IT GO OFF OF THE END?
055136  001003                          BNE     2$              ;;BRANCH IF NO
055140  012737  054204  054202          MOV     #$TKQSRT,$TKQOUT ;;RESET THE POINTER
055146  000002                  2$:     RTI                     ;;RETURN
                                        ;;**********************************************************************
                                        ;*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

055150  010346                  $RDLIN: MOV     R3,-(SP)        ;;SAVE R3
055152  012703  055256          1$:     MOV     #$TTYIN,R3      ;;GET ADDRESS
055156  022703  055267          2$:     CMP     #$TTYIN+9.,R3   ;;BUFFER FULL?
055162  101405                          BLOS    4$              ;;BR IF YES
055164  104410                          RDCHR                   ;;GO READ ONE CHARACTER FROM THE TTY
055166  112613                          MOVB    (SP)+,(R3)      ;;GET CHARACTER
055170  122713  000177          10$:    CMPB    #177,(R3)       ;;IS IT A RUBOUT
055174  001003                          BNE     3$              ;;SKIP IF NOT
055176  104401  001222          4$:     TYPE    ,$QUES          ;;TYPE A '?'
055202  000763                          BR      1$              ;;CLEAR THE BUFFER AND LOOP
055204  111337  055254          3$:     MOVB    (R3),9$         ;;ECHO THE CHARACTER
055210  104401  055254                  TYPE    ,9$
055214  122723  000015                  CMPB    #15,(R3)+       ;;CHECK FOR RETURN
055220  001356                          BNE     2$              ;;LOOP IF NOT RETURN
055222  105063  177777                  CLRB    -1(R3)          ;;CLEAR RETURN (THE 15)
055226  104401  001224                  TYPE    ,$LF            ;;TYPE A LINE FEED
055232  012603                          MOV     (SP)+,R3        ;;RESTORE R3
055234  011646                          MOV     (SP),-(SP)      ;;ADJUST THE STACK AND PUT ADDRESS OF THE
055236  016666  000004  000002          MOV     4(SP),2(SP)     ;;      FIRST ASCII CHARACTER ON IT
055244  012766  055256  000004          MOV     #$TTYIN,4(SP)
055252  000002                          RTI                     ;;RETURN
055254     000                  9$:     .BYTE   0               ;;STORAGE FOR ASCII CHAR. TO TYPE
055255     000                          .BYTE   0               ;;TERMINATOR
055256                          $TTYIN: .BLKB   9.              ;;RESERVE 9. BYTES FOR TTY INPUT
055267     136     103     015  $CNTLC: .ASCIZ  /^C/<15><12>    ;;CONTROL "C"
055274     136     125     015  $CNTLU: .ASCIZ  /^U/<15><12>    ;;CONTROL "U"
055301     136     107     015  $CNTLG: .ASCIZ  /^G/<15><12>    ;;CONTROL "G"
```

```
055306  015  012  123  $MSWR:  .ASCIZ  <15><12>/SWR = /
055317  040  040  116  $MNEW:  .ASCIZ  /  NEW = /
```

```
                                .SBTTL  READ AN OCTAL NUMBER FROM THE TTY

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

055330  011646                  $RDOCT: MOV     (SP),-(SP)      ;;PROVIDE SPACE FOR THE
055332  016666  000004  000002          MOV     4(SP),2(SP)     ;;INPUT NUMBER
055340  010046                          MOV     R0,-(SP)        ;;PUSH R0 ON STACK
055342  010146                          MOV     R1,-(SP)        ;;PUSH R1 ON STACK
055344  010246                          MOV     R2,-(SP)        ;;PUSH R2 ON STACK
055346  104411                  1$:     RDLIN                   ;;READ AN ASCIZ LINE
055350  012600                          MOV     (SP)+,R0        ;;GET ADDRESS OF 1ST CHARACTER
055352  010037  055456                  MOV     R0,5$           ;;AND SAVE IT
055356  005001                          CLR     R1              ;;CLEAR DATA WORD
055360  005002                          CLR     R2
055362  112046                  2$:     MOVB    (R0)+,-(SP)     ;;PICKUP THIS CHARACTER
055364  001420                          BEQ     3$              ;;IF ZERO GET OUT
055366  122716  000060                  CMPB    #'0,(SP)        ;;MAKE SURE THIS CHARACTER
055372  003026                          BGT     4$              ;;IS AN OCTAL DIGIT
055374  122716  000067                  CMPB    #'7,(SP)
055400  002423                          BLT     4$
055402  006301                          ASL     R1              ;;*2
055404  006102                          ROL     R2
055406  006301                          ASL     R1              ;;*4
055410  006102                          ROL     R2
055412  006301                          ASL     R1              ;;*8
055414  006102                          ROL     R2
055416  042716  177770                  BIC     #^C7,(SP)       ;;STRIP THE ASCII JUNK
055422  062601                          ADD     (SP)+,R1        ;;ADD IN THIS DIGIT
055424  000756                          BR      2$              ;;LOOP
055426  005726                  3$:     TST     (SP)+           ;;CLEAN TERMINATOR FROM STACK
055430  010166  000012                  MOV     R1,12(SP)       ;;SAVE THE RESULT
055434  010237  055466                  MOV     R2,$HIOCT
055440  012602                          MOV     (SP)+,R2        ;;POP STACK INTO R2
055442  012601                          MOV     (SP)+,R1        ;;POP STACK INTO R1
055444  012600                          MOV     (SP)+,R0        ;;POP STACK INTO R0
055446  000002                          RTI                     ;;RETURN
055450  005726                  4$:     TST     (SP)+           ;;CLEAN PARTIAL FROM STACK
055452  105010                          CLRB    (R0)            ;;SET A TERMINATOR
055454  104401                          TYPE                    ;;TYPE UP THRU THE BAD CHAR.
055456  000000                  5$:     .WORD   0
055460  104401  001222                  TYPE    ,$QUES          ;;"?" "CR" & "LF"
055464  000730                          BR      1$              ;;TRY AGAIN
055466  000000                  $HIOCT: .WORD   0               ;;HIGH ORDER BITS GO HERE
```

```
                                .SBTTL  ERROR HANDLER ROUTINE

                                ;;**********************************************************************
                                ;*THIS ROUTINE WILL INCREMENT THE ERROR FLAG AND THE ERROR COUNT,
                                ;*SAVE THE ERROR ITEM NUMBER AND THE ADDRESS OF THE ERROR CALL
                                ;*AND GO TO $ERRTYP ON ERROR
                                ;*THE SWITCH OPTIONS PROVIDED BY THIS ROUTINE ARE:
                                ;*SW15=1         HALT ON ERROR
                                ;*SW13=1         INHIBIT ERROR TYPEOUTS
                                ;*SW10=1         BELL ON ERROR
                                ;*SW09=1         LOOP ON ERROR
                                ;*CALL
                                ;*      ERROR   N       ;;ERROR=EMT AND N=ERROR ITEM NUMBER

055470  105037  056030          $ERROR: CLRB    IBSAVE          ;;CLEAR THE ITEM BYTE SAVE LOCATION
055474  104407                          CKSWR                   ;;TEST FOR CHANGE IN SOFT-SWR
055476  012737  177777  001406          MOV     #-1,ERFLG$      ;SET ERROR FLAG
055504                          REGSA1:
055504  105237  001103          7$:     INCB    $ERFLG          ;;SET THE ERROR FLAG
055510  001775                          BEQ     7$              ;;DON'T LET THE FLAG GO TO ZERO
055512  013777  001102  123422          MOV     $TSTNM,@DISPLAY ;;DISPLAY TEST NUMBER AND ERROR FLAG
055520  032777  002000  123412          BIT     #BIT10,@SWR     ;;BELL ON ERROR?
055526  001402                          BEQ     1$              ;;NO - SKIP
055530  104401  001216                  TYPE    ,$BELL          ;;RING BELL
055534  005237  001112          1$:     INC     $ERTTL          ;;COUNT THE NUMBER OF ERRORS
055540  011637  001116                  MOV     (SP),$ERRPC     ;;GET ADDRESS OF ERROR INSTRUCTION
055544  162737  000002  001116          SUB     #2,$ERRPC
055552  117737  123340  001114          MOVB    @$ERRPC,$ITEMB  ;;STRIP AND SAVE THE ERROR ITEM CODE
055560  032777  001000  123352          BIT     #BIT09,@SWR     ;;SEE IF LOOP ON ERROR IS SET
055566  001060                          BNE     1004$           ;;BRANCH AROUND ROUTINE IF SO
055570  122737  000177  001114          CMPB    #177,$ITEMB     ;;SEE IF THIS IS THE POWER FAIL CALL
055576  001454                          BEQ     1004$           ;;BRANCH AROUND ROUTINE IF IT IS
055600  105737  056030                  TSTB    IBSAVE          ;;SEE IF THIS IS THE 2ND ERROR CALL IN THIS ROUTINE
055604  001047                          BNE     1003$           ;;BRANCH IF SO
055606  022737  177777  056026          CMP     #-1,CPSAVE      ;;SEE IF CPSAVE HAS CPU ERR REG TIMEOUT INDICATION
055614  001445                          BEQ     1004$           ;;BRANCH IF SO
055616  013746  000004                  MOV     ERRVEC,-(SP)    ;;SAVE CONTENTS OF ERROR VECTOR
055622  012737  055640  000004          MOV     #1000$,ERRVEC   ;;SETUP 'TRAP' RETURN ADDRESS
055630  013737  177766  056026          MOV     177766,CPSAVE   ;;MOVE CPU ERROR REGISTER TO CPSAVE FOR TEST
055636  000406                          BR      1001$
055640  012737  177777  056026  1000$:  MOV     #-1,CPSAVE      ;;SET CPU ERROR REGISTER TIMEOUT INDICATOR
055646  012716  055654                  MOV     #1001$,(SP)     ;;SETUP RETURN ADDRESS
055652  000002                          RTI
055654  012637  000004          1001$:  MOV     (SP)+,ERRVEC    ;;RESTORE CONTENTS OF ERROR VECTOR

055660  022737  177777  056026  1002$:  CMP     #-1,CPSAVE      ;;SEE IF CPSAVE HAS CPU ERR REG TIMEOUT INDICATION
055666  001420                          BEQ     1004$           ;;BRANCH IF SO
055670  032737  000001  056026          BIT     #BIT00,CPSAVE   ;;SEE IF POWER MONITOR BIT IS SET IN CPU ERR REG
055676  001414                          BEQ     1004$           ;;BRANCH IF OK
055700  042737  000001  177766          BIC     #BIT00,177766   ;;CLEAR THE BIT FOUND SET
055706  113737  001114  056030          MOVB    $ITEMB,IBSAVE   ;;MAKE IBSAVE NON-ZERO FOR DUAL ERROR CALL
055714  112737  000177  001114          MOVB    #177,$ITEMB     ;;SET $ITEMB TO SPECIAL POWER FAIL POINTER
055722  000402                          BR      1004$           ;;BRANCH OVER IBSAVE CLEARING

055724  105037  056030          1003$:  CLRB    IBSAVE          ;;CLEAR IBSAVE SO 2ND TIME THROUGH EXITS
055730                          1004$:
055730  032777  020000  123202          BIT     #BIT13,@SWR     ;;SKIP TYPEOUT IF SET
```

ERROR HANDLER ROUTINE

```
055736  001004                          BNE     20$             ;;SKIP TYPEOUTS
055740  004737  056032                  JSR     PC,$ERRTYP      ;;GO TO USER ERROR ROUTINE
055744  104401  001223                  TYPE    ,$CRLF
055750                          20$:
055750  105737  056030          2$:     TSTB    IBSAVE          ;;SEE IF IBSAVE IS LOADED
055754  001005                          BNE     3$              ;;BRANCH IF NOT - NO HALT ON PWR MON BIT ERROR
055756  005777  123156                  TST     @SWR            ;;HALT ON ERROR
055762  100002                          BPL     3$              ;;SKIP IF CONTINUE
055764  000000                          HALT                    ;;HALT ON ERROR!
055766  104407                          CKSWR                   ;;TEST FOR CHANGE IN SOFT-SWR
055770                          3$:
055770  032777  001000  123142          BIT     #BIT09,@SWR     ;;LOOP ON ERROR SWITCH SET?
055776  001402                          BEQ     4$              ;;BR IF NO
056000  013716  001110                  MOV     $LPERR,(SP)     ;;FUDGE RETURN FOR LOOPING
056004  005737  001214          4$:     TST     $ESCAPE         ;;CHECK FOR AN ESCAPE ADDRESS
056010  001402                          BEQ     5$              ;;BR IF NONE
056012  013716  001214                  MOV     $ESCAPE,(SP)    ;;FUDGE RETURN ADDRESS FOR ESCAPE
056016                          5$:
056016  105737  056030                  TSTB    IBSAVE          ;;SEE IF ITEM BYTE SAVE LOCATION HAS AN ERROR CALL
056022  001230                          BNE     7$              ;;BRANCH BACK TO CALL ORIGINAL ERROR
056024  000002                          RTI                     ;;RETURN
056026  000000                  CPSAVE: .WORD   0               ;;LOCATION TO SAVE CPU ERROR REG CONTENTS
056030  000000                  IBSAVE: .WORD   0               ;;LOCATION TO SAVE ITEM BYTE
```

ERROR MESSAGE TYPEOUT ROUTINE

```
                                .SBTTL  ERROR MESSAGE TYPEOUT ROUTINE

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

056032                          $ERRTYP:
056032  104401  001223                TYPE    ,$CRLF          ;;"CARRIAGE RETURN" & "LINE FEED"
056036  010046                        MOV     R0,-(SP)        ;;SAVE R0
056040  005000                        CLR     R0              ;;PICKUP THE ITEM INDEX
056042  153700  001114                BISB    @#$ITEMB,R0
056046  001004                        BNE     1$              ;;IF ITEM NUMBER IS ZERO, JUST
                                                                ;;TYPE THE PC OF THE ERROR
056050  013746  001116                MOV     $ERRPC,-(SP)    ;;SAVE $ERRPC FOR TYPEOUT
                                                                ;;ERROR ADDRESS
056054  104402                        TYPOC                   ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
056056  000456                        BR      10$             ;;GET OUT
056060  122700  000177          1$:   CMPB    #177,R0         ;;SEE IF THIS ERROR CALL IS SPECIAL POWER FAIL CALL
056064  001006                        BNE     1000$           ;;BRANCH IF NOT
056066  113737  001102  056370        MOVB    $TSTNM,PFTSTN   ;;GET TEST NUMBER
056074  012700  056230                MOV     #PFECH,R0       ;;MOVE POWER FAIL ERROR CALL TABLE TO R0
056100  000406                        BR      1001$           ;;BRANCH TO CALL ERROR
056102  005300                1000$:  DEC     R0              ;;ADJUST THE INDEX SO THAT IT WILL
056104  006300                        ASL     R0              ;;      WORK FOR THE ERROR TABLE
056106  006300                        ASL     R0
056110  006300                        ASL     R0
056112  062700  003630                ADD     #$ERRTB,R0      ;;FORM TABLE POINTER
056116  012037  056126        1001$:  MOV     (R0)+,2$        ;;PICKUP "ERROR MESSAGE" POINTER
056122  001404                        BEQ     3$              ;;SKIP TYPEOUT IF NO POINTER
056124  104401                        TYPE                    ;;TYPE THE "ERROR MESSAGE"
056126  000000                2$:     .WORD   0               ;;"ERROR MESSAGE" POINTER GOES HERE
056130  104401  001223                TYPE    ,$CRLF          ;;"CARRIAGE RETURN" & "LINE FEED"
056134  012037  056144        3$:     MOV     (R0)+,4$        ;;PICKUP "DATA HEADER" POINTER
056140  001404                        BEQ     5$              ;;SKIP TYPEOUT IF 0
056142  104401                        TYPE                    ;;TYPE THE "DATA HEADER"
056144  000000                4$:     .WORD   0               ;;"DATA HEADER" POINTER GOES HERE
056146  104401  001223                TYPE    ,$CRLF          ;;"CARRIAGE RETURN" & "LINE FEED"
056152  010146                5$:     MOV     R1,-(SP)        ;;SAVE R1
056154  012001                        MOV     (R0)+,R1        ;;PICKUP "DATA TABLE" POINTER
056156  001415                        BEQ     9$              ;;BR IF NO DATA TO BE TYPED
056160  012000                        MOV     (R0)+,R0        ;;PICKUP "DATA FORMAT" POINTER
056162  105720                6$:     TSTB    (R0)+           ;;"OCTAL" OR "DECIMAL"
056164  001003                        BNE     7$              ;;BR IF DECIMAL
056166  013146                        MOV     @(R1)+,-(SP)    ;;SAVE @(R1)+ FOR TYPEOUT
056170  104402                        TYPOC                   ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
056172  000402                        BR      8$
056174                          7$:
056174  013146                        MOV     @(R1)+,-(SP)    ;;SAVE @(R1)+ FOR TYPEOUT
056176  104405                        TYPDS                   ;;GO TYPE--DECIMAL ASCII WITH SIGN
056200  005711                8$:     TST     (R1)            ;;IS THERE ANOTHER NUMBER?
056202  001403                        BEQ     9$              ;;BR IF NO
056204  104401  056224                TYPE    ,11$            ;;TYPE TWO(2) SPACES
056210  000764                        BR      6$              ;;LOOP

056212  012601                9$:     MOV     (SP)+,R1        ;;RESTORE R1
056214  012600                10$:    MOV     (SP)+,R0        ;;RESTORE R0
```

ERROR MESSAGE TYPEOUT ROUTINE

```
056216  104401  001223                  TYPE    ,$CRLF          ;;"CARRIAGE RETURN" & "LINE FEED"
056222  000207                          RTS     PC              ;;RETURN
056224     040     040     000  11$:     .ASCIZ  /  /            ;;TWO(2) SPACES
                                        .EVEN
056230  056240  056322  056354  PFECH:  PFECH1,PFECH2,PFECH3,PFECH4 ;;WORDS DEFINING TABLES BELOW
056240     120     117     127  PFECH1: .ASCIZ  ?POWER MONITOR BIT IN CPU ERROR REGISTER FOUND SET?
056322     124     105     123  PFECH2: .ASCIZ  ?TESTNO  ERR PC  CPUERREG?
                                        .EVEN
056354  056370  001116  056026  PFECH3: .WORD   PFTSTN,$ERRPC,CPSAVE,0
056364     000     000     000  PFECH4: .BYTE   0,0,0,0
056370  000000                  PFTSTN: .WORD   0               ;;CONTAINS TEST NUMBER FOR PF BIT ERROR
```

```
                                .SBTTL  BINARY TO OCTAL (ASCII) AND TYPE

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

056372  017646  000000        $TYPOS: MOV     @(SP),-(SP)     ;;PICKUP THE MODE
056376  116637  000001 056615          MOVB    1(SP),$OFILL    ;;LOAD ZERO FILL SWITCH
056404  112637  056617                 MOVB    (SP)+,$OMODE+1  ;;NUMBER OF DIGITS TO TYPE
056410  062716  000002                 ADD     #2,(SP)         ;;ADJUST RETURN ADDRESS
056414  000406                         BR      $TYPON
056416  112737  000001 056615  $TYPOC: MOVB    #1,$OFILL       ;;SET THE ZERO FILL SWITCH
056424  112737  000006 056617          MOVB    #6,$OMODE+1     ;;SET FOR SIX(6) DIGITS
056432  112737  000005 056614  $TYPON: MOVB    #5,$OCNT        ;;SET THE ITERATION COUNT
056440  010346                         MOV     R3,-(SP)        ;;SAVE R3
056442  010446                         MOV     R4,-(SP)        ;;SAVE R4
056444  010546                         MOV     R5,-(SP)        ;;SAVE R5
056446  113704  056617                 MOVB    $OMODE+1,R4     ;;GET THE NUMBER OF DIGITS TO TYPE
056452  005404                         NEG     R4
056454  062704  000006                 ADD     #6,R4           ;;SUBTRACT IT FOR MAX. ALLOWED
056460  110437  056616                 MOVB    R4,$OMODE       ;;SAVE IT FOR USE
056464  113704  056615                 MOVB    $OFILL,R4       ;;GET THE ZERO FILL SWITCH
056470  016605  000012                 MOV     12(SP),R5       ;;PICKUP THE INPUT NUMBER
056474  005003                         CLR     R3              ;;CLEAR THE OUTPUT WORD
056476  006105                  1$:    ROL     R5              ;;ROTATE MSB INTO "C"
056500  000404                         BR      3$              ;;GO DO MSB
056502  006105                  2$:    ROL     R5              ;;FORM THIS DIGIT
056504  006105                         ROL     R5
056506  006105                         ROL     R5
056510  010503                         MOV     R5,R3
056512  006103                  3$:    ROL     R3              ;;GET LSB OF THIS DIGIT
056514  105337  056616                 DECB    $OMODE          ;;TYPE THIS DIGIT?
056520  100016                         BPL     7$              ;;BR IF NO
056522  042703  177770                 BIC     #177770,R3      ;;GET RID OF JUNK
056526  001002                         BNE     4$              ;;TEST FOR 0
056530  005704                         TST     R4              ;;SUPPRESS THIS 0?
056532  001403                         BEQ     5$              ;;BR IF YES
056534  005204                  4$:    INC     R4              ;;DON'T SUPPRESS ANYMORE 0'S
```

```
056536  052703  000060            BIS     #'0,R3          ;;MAKE THIS DIGIT ASCII
056542  052703  000040      5$:   BIS     #' ,R3          ;;MAKE ASCII IF NOT ALREADY
056546  110337  056612            MOVB    R3,8$           ;;SAVE FOR TYPING
056552  104401  056612            TYPE    ,8$             ;;GO TYPE THIS DIGIT
056556  105337  056614      7$:   DECB    $OCNT           ;;COUNT BY 1
056562  003347                    BGT     2$              ;;BR IF MORE TO DO
056564  002402                    BLT     6$              ;;BR IF DONE
056566  005204                    INC     R4              ;;INSURE LAST DIGIT ISN'T A BLANK
056570  000744                    BR      2$              ;;GO DO THE LAST DIGIT
056572  012605      6$:           MOV     (SP)+,R5        ;;RESTORE R5
056574  012604                    MOV     (SP)+,R4        ;;RESTORE R4
056576  012603                    MOV     (SP)+,R3        ;;RESTORE R3
056600  016666  000002  000004    MOV     2(SP),4(SP)     ;;SET THE STACK FOR RETURNING
056606  012616                    MOV     (SP)+,(SP)
056610  000002                    RTI                     ;;RETURN
056612     000      8$:           .BYTE   0               ;;STORAGE FOR ASCII DIGIT
056613     000                    .BYTE   0               ;;TERMINATOR FOR TYPE ROUTINE
056614     000      $OCNT:        .BYTE   0               ;;OCTAL DIGIT COUNTER
056615     000      $OFILL:       .BYTE   0               ;;ZERO FILL SWITCH
056616  000000      $OMODE:       .WORD   0               ;;NUMBER OF DIGITS TO TYPE
```

```
                                .SBTTL  TRAP DECODER

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

056620  010046                  $TRAP:  MOV     R0,-(SP)        ;;SAVE R0
056622  016600  000002                  MOV     2(SP),R0        ;;GET TRAP ADDRESS
056626  005740                          TST     -(R0)           ;;BACKUP BY 2
056630  111000                          MOVB    (R0),R0         ;;GET RIGHT BYTE OF TRAP
056632  006300                          ASL     R0              ;;POSITION FOR INDEXING
056634  016000  056654                  MOV     $TRPAD(R0),R0   ;;INDEX TO TABLE
056640  000200                          RTS     R0              ;;GO TO ROUTINE


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

056642  011646                  $TRAP2: MOV     (SP),-(SP)      ;;MOVE THE PC DOWN
056644  016666  000004  000002          MOV     4(SP),2(SP)     ;;MOVE THE PSW DOWN
056652  000002                          RTI                     ;;RESTORE THE PSW

                                .SBTTL  TRAP TABLE

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

                                ;       ROUTINE
                                ;       -------
056654  056642                  $TRPAD: .WORD   $TRAP2
056656  053704                          $TYPE   ;;CALL=TYPE     TRAP+1(104401)  TTY TYPEOUT ROUTINE
056660  056416                          $TYPOC  ;;CALL=TYPOC    TRAP+2(104402)  TYPE OCTAL NUMBER (WITH LEADING ZEROS)
056662  056372                          $TYPOS  ;;CALL=TYPOS    TRAP+3(104403)  TYPE OCTAL NUMBER (NO LEADING ZEROS)
056664  056432                          $TYPON  ;;CALL=TYPON    TRAP+4(104404)  TYPE OCTAL NUMBER (AS PER LAST CALL)
056666  053460                          $TYPDS  ;;CALL=TYPDS    TRAP+5(104405)  TYPE DECIMAL NUMBER (WITH SIGN)

056670  054606                          $GTSWR  ;;CALL=GTSWR    TRAP+6(104406)  GET SOFT-SWR SETTING

056672  054516                          $CKSWR  ;;CALL=CKSWR    TRAP+7(104407)  TEST FOR CHANGE IN SOFT-SWR
056674  055060                          $RDCHR  ;;CALL=RDCHR    TRAP+10(104410) TTY TYPEIN CHARACTER ROUTINE
056676  055150                          $RDLIN  ;;CALL=RDLIN    TRAP+11(104411) TTY TYPEIN STRING ROUTINE
056700  055330                          $RDOCT  ;;CALL=RDOCT    TRAP+12(104412) READ AN OCTAL NUMBER FROM TTY
2 056702  041442                        T.SCOP  ;;CALL=SCOP1    TRAP+13(104413) MY LOCAL SCOPES
3 056704  041514                        CHECKT  ;;CALL=CHECKD   TRAP+14(104414) CHECK DVA,RDY,DPR,DRY
4 056706  042032                        WAIT.T  ;;CALL=WAT      TRAP+15(104415) WAIT LOOP
```

```
     1                                  .SBTTL  POWER DOWN AND UP ROUTINES

                                        ;;*****************************************************************
                                        ;POWER DOWN ROUTINE
056710  012737  057054  000024  $PWRDN: MOV     #$ILLUP,@#PWRVEC ;;SET FOR FAST UP
056716  012737  000340  000026         MOV     #340,@#PWRVEC+2 ;;PRIO:7
056724  010046                       MOV     R0,-(SP)        ;;PUSH R0 ON STACK
056726  010146                       MOV     R1,-(SP)        ;;PUSH R1 ON STACK
056730  010246                       MOV     R2,-(SP)        ;;PUSH R2 ON STACK
056732  010346                       MOV     R3,-(SP)        ;;PUSH R3 ON STACK
056734  010446                       MOV     R4,-(SP)        ;;PUSH R4 ON STACK
056736  010546                       MOV     R5,-(SP)        ;;PUSH R5 ON STACK
056740  017746  122174               MOV     @SWR,-(SP)      ;;PUSH @SWR ON STACK
056744  010637  057060               MOV     SP,$SAVR6       ;;SAVE SP
056750  012737  056762  000024       MOV     #$PWRUP,@#PWRVEC ;;SET UP VECTOR
056756  000000                       HALT
056760  000776                       BR      .-2             ;;HANG UP

                                        ;;*****************************************************************
                                        ;POWER UP ROUTINE
056762  012737  057054  000024  $PWRUP: MOV     #$ILLUP,@#PWRVEC ;;SET FOR FAST DOWN
056770  013706  057060               MOV     $SAVR6,SP       ;;GET SP
056774  005037  057060               CLR     $SAVR6          ;;WAIT LOOP FOR THE TTY
057000  005237  057060         1$:   INC     $SAVR6          ;;WAIT FOR THE INC
057004  001375                       BNE     1$              ;;OF  WORD
057006  012677  122126               MOV     (SP)+,@SWR      ;;POP STACK INTO @SWR
057012  012605                       MOV     (SP)+,R5        ;;POP STACK INTO R5
057014  012604                       MOV     (SP)+,R4        ;;POP STACK INTO R4
057016  012603                       MOV     (SP)+,R3        ;;POP STACK INTO R3
057020  012602                       MOV     (SP)+,R2        ;;POP STACK INTO R2
057022  012601                       MOV     (SP)+,R1        ;;POP STACK INTO R1
057024  012600                       MOV     (SP)+,R0        ;;POP STACK INTO R0
057026  012737  056710  000024       MOV     #$PWRDN,@#PWRVEC ;;SET UP THE POWER DOWN VECTOR
057034  012737  000340  000026       MOV     #340,@#PWRVEC+2 ;;PRIO:7
057042  104401                       TYPE                    ;REPORT THE POWER FAILURE
057044  057062               $PWRMG: .WORD   $POWER          ;;POWER FAIL MESSAGE POINTER
057046  012716                       MOV     (PC)+,(SP)      ;;RESTART AT BEGIN
057050  004310               $PWRAD: .WORD   BEGIN           ;;RESTART ADDRESS
057052  000002                       RTI
057054  000000               $ILLUP: HALT                    ;;THE POWER UP SEQUENCE WAS STARTED
057056  000776                       BR      .-2             ;; BEFORE THE POWER DOWN WAS COMPLETE
057060  000000               $SAVR6: 0                       ;;PUT THE SP HERE
057062     015      012      120  $POWER: .ASCIZ  <15><12>''POWER''
                                        .EVEN
```

```
                                        .SBTTL  ERROR AND MESSAGE TABLE CONDIMENTS
     2
     3 057072   127   122   117  EM1:    .ASCIZ  /WRONG DATA IN READING OR WRITING HARDWARE REGISTER/
     4 057155   105   122   122  EM2:    .ASCIZ  /ERROR ON  DATA COMMAND/
     5 057204   105   122   122  EM6:    .ASCIZ  /ERROR ON WRITE HEADER AND DATA/
     6 057243   103   117   116  EM11:   .ASCIZ  /CONTROLLER OR DRIVE STATUS/
     7 057276   122   105   107  EM14:   .ASCIZ  /REGISTER FAILED/
     8 057316   116   117   116  EM15:   .ASCIZ  /NON EXISTENT REGISTER, PROGRAM ABORTED./
     9 057366   127   101   111  EM16:   .ASCIZ  /WAIT LOOP FAILED/
    10 057407   127   122   111  EM17:   .ASCIZ  /WRITE CHECK FAILING/
    11 057433   122   105   107  EM20:   .ASCIZ  /REGISTER FAILING/
    12 057454   111   116   124  EM21:   .ASCIZ  /INTERRUPT FAILING/
    13 057476   105   122   122  EM22:   .ASCII  /ERROR ON DRIVES PRESENT -/<CRLF>
    14 057530   124   110   105          .ASCII  /THE UNIT NO'S FOUND BY SETTING RHAS USING RHER1/<CRLF>
    15 057610   050   124   064          .ASCII  /(T4) DO NOT AGREE WITH THE UNIT NO'S FOUND/<CRLF>
    16 057663   102   131   040          .ASCII  /BY LOOKING FOR 'NED' = 0 IN RHCS2 (BIT #12)/<CRLF><LF>
    17 057740   116   117   124          .ASCII  /NOTE: ON DUAL PORT SYSTEM, A DRIVE ON OTHER PORT WILL /<CRLF>
    18 060027   116   117   124          .ASCII  /NOT GIVE 'NED', BUT WILL GIVE RHAS RESPONSES/<CRLF>
    19 060104   110   105   116          .ASCIZ  /HENCE THERE WILL BE AN EXTRA DRIVE/
    20 060147   114   117   117  EM24:   .ASCIZ  /LOOK AHEAD REGISTER AT THE BEGINNING OF SECTOR IS IN ERROR/
    21 060242   114   117   117  EM25:   .ASCIZ  /LOOK AHEAD REGISTER IS IN ERROR/
    22 060302   103   125   122  EM30:   .ASCII  /CURRENT CYLINDER DOES NOT MATCH DESIRED CYLINDER REGISTER/<CRLF>
    23 060374   101   106   124          .ASCIZ  /AFTER A SEEK AND INIT/
    24 060422   105   103   103  EM31:   .ASCII  /ECC GENERATED IS INCORRECT/<CRLF>
    25 060455   105   126   105          .ASCIZ  /EVERY WORD ON THIS SECTOR IS THAT GIVEN IN 'DATA USED'/
    26 060544   117   116   040  EM32:   .ASCII  /ON READ COMMAND, AFTER DATA AND ECC HAVE BEEN READ,/<CRLF>
    27 060630   105   103   103          .ASCII  /ECC REGISTERS OR RHER1 ARE IN ERROR/<CRLF>
    28 060674   117   116   114          .ASCII  /ONLY LOWER 11 BITS OF PATTERN REG. CAN BE READ/<CRLF>
    29 060753   124   110   111          .ASCIZ  /THIS SHOULD MATCH LOWER 11 BITS OF GOOD ECC1/
    30 061030   110   111   107  EM33:   .ASCIZ  /HIGH COUNT BIT NOT SET AFTER 38859 CLOCKS/
    31 061102   132   105   122  EM34:   .ASCIZ  /ZERO DETECT BIT NOT HIGH WHEN 32 BIT ECC REG. HAS 21 ZEROS/
    32 061175   120   117   123  EM35:   .ASCII  /POSITION REGISTER OR 11 BITS OF PATTERN REGISTER INCORRECT/<CRLF>
    33 061270   114   117   127          .ASCII  /LOWER 11 BITS OF PATTERN REGISTER SHOULD MATCH LOWER/<CRLF>
    34 061355   061   061   040          .ASCII  /11 BITS OF GOOD ECC1/<CRLF>
    35 061402   104   101   124          .ASCIZ  /DAT ENVLOP GOOD POSITION AND N-CODE ZEROS ARE IN OCTAL/
    36 061471   117   116   040  EM36:   .ASCIZ  /ON READ COMMAND WITH NON-CORRECTABLE ERROR 'DCK' AND 'ECH' SHOULD BE SET/
    37 061602   120   122   117  EM37:   .ASCII  /PROGRAM ERROR BIT #10 IN RHCS2 DID NOT SET/<CRLF>
    38 061655   111   106   040          .ASCIZ  /IF POSITION REGISTER =10040 OR 10041, IT IS GOOD/
    39
    40 061736   122   110   127  EM40:   .ASCII  /RHWC DID NOT = 0 UPON COMPLETION OF READ/<CRLF>
    41 062007   117   122   040          .ASCIZ  /OR WRITE HEADER AND DATA/
```

ERROR AND MESSAGE TABLE CONDIMENTS

```
 1 062040      200      101      114  PREAMB: .ASCII  <CRLF>/ALL DCL'S UNDER TEST MUST BE LOCKED ON CORRECT PORT./
 2 062125      200      111      106          .ASCII  <CRLF>/IF CHANGES ARE REQUIRED ON PORT SWITCH, THEN A CYCLE/
 3 062212      200      125      120          .ASCII  <CRLF>/UP SEQUENCE IS REQUIRED FOR STROBING THE PORT SELECT/
 4 062277      200      106      114          .ASCII  <CRLF>/FLOP./
 5 062305      200                            .ASCII  <CRLF>
 6 062306      200      101      114          .ASCII  <CRLF>/ALL  DCL'S NOT  UNDER TEST  MUST BE  SWITCHED OFF OR/
 7 062373      200      114      117          .ASCIZ  <CRLF>/LOCKED ON THE OTHER PORT./
 8
 9 062426      200      116      117  NDRVAS: .ASCII  <CRLF>/NO DRIVES PRESENT, RHAS=0/
10 062460      200                            .ASCII  <CRLF>
11 062461      200      127      122          .ASCII  <CRLF>/WRITTING ONES  INTO RHER1  FOR ALL UNIT NUMBERS DOES/
12 062546      200      116      117          .ASCII  <CRLF>/NOT SET ANY BIT IN RHAS, SO ABORT PROGRAM./
13 062621      200                            .ASCII  <CRLF>
14 062622      200      124      117          .ASCII  <CRLF>/TO LOOP ON THIS TEST WITHOUT PRINTOUT,  SET SWITCHES/
15 062707      200      061      063          .ASCIZ  <CRLF>/13, 8 AND 2./<CRLF>
16
17 062726      106      101      124  CPHALT: .ASCII  /FATAL ERROR - SEE DOCUMENT LISTING/<CRLF>
18 062771      040      200      207          .ASCII  / /<CRLF><207><377><377><207><377><377><207><377><377>
19 063004      124      110      105          .ASCII  /THE CONTROLLER OR DEVICE HAS GONE OFFLINE, LOST/<CRLF>
20 063064      047      122      105          .ASCII  /'READY', BECOME UNAVAILABLE, OR HAS STATUS BITS/<CRLF>
21 063144      127      110      111          .ASCIZ  /WHICH CANNOT BE CLEARED/
22
23 063174      040               BLNKS8: .ASCII  / /
24 063175      040               BLNKS7: .ASCII  / /
25 063176      040               BLNKS6: .ASCII  / /
26 063177      040               BLNKS5: .ASCII  / /
27 063200      040               BLNKS4: .ASCII  / /
28 063201      040               BLNKS3: .ASCII  / /
29 063202      040               BLNKS2: .ASCII  / /
30 063203      040      000      BLNKS1: .ASCIZ  / /
```

ERROR AND MESSAGE TABLE CONDIMENTS

```
 1 063205  120  103  040  DH1:   .ASCII  /PC  TEST  REG.   GOOD   RECEIVED/<CRLF>
 2 063256  040  040  040         .ASCIZ  /    NO    ADDR.  DATA   DATA   /
 3 063327  120  103  040  DH2:   .ASCII  /PC  TEST  WORD   GOOD   BAD    /<CRLF>
 4 063400  040  040  040         .ASCIZ  /    NO    NO     DATA   DATA   /
 5 063451  120  103  040  DH11:  .ASCII  /PC  TEST  FAILING CONT.  CONT.  CONT.  CONT.  /<CRLF>
 6 063542  040  040  040         .ASCIZ  /    NO    REG ADR RHCS1  RHCS2  RHDS1  RHER1/
 7 063630  120  103  040  DH14:  .ASCII  /PC  TEST  FAILING CONT.  CONT.  CONT.  CONT.  CONT.  /<CRLF>
 8 063731  040  040  040         .ASCIZ  /    NO    REG ADR BAD REG RHCS1  RHCS2  RHDS1  RHER1/
 9 064027  120  103  040  DH15:  .ASCIZ  /PC  TEST  REG ADR /
10 064060  120  103  040  DH16:  .ASCII  /PC  TEST  WAIT   BIT    REG    REG    /<CRLF>
11 064141  040  040  040         .ASCIZ  /    NO    PC     WANTED ADDRESS CONT.  /
12 064222  120  103  040  DH17:  .ASCII  /PC  TEST  CONT.  CONT.  CONT.  CONT.  CONT.  /<CRLF>
13 064313  040  040  040         .ASCIZ  /    NO    RHBA   RHDB   RHWC   RHCS1  RHCS2  /
14 064404  120  103  040  DH20:  .ASCII  /PC  TEST  CONT   CONT   CONT   CONT   CONT   /<CRLF>
15 064475  040  040  040         .ASCIZ  /    NO    RHER1  RHER2  RHER3  RHAS   RHDS1  /
16 064566  120  103  040  DH21:  .ASCII  /PC  TEST  CONT   CONT   CONT   /<CRLF>
17 064637  040  040  040         .ASCIZ  /    NO    RHCS1  RHAS   RHDS1/
18 064705  120  103  040  DH22:  .ASCII  /PC  TEST     RHAS     RHCS2/<CRLF>
19 064746  040  040  040         .ASCIZ  /    NO     UNITS    UNITS/
20 065007  120  103  040  DH24:  .ASCII  /PC  TEST  RHDST  BAD    GOOD   SECTOR SECTOR /<CRLF>
21 065100  040  040  040         .ASCIZ  /    NO    CONT.  RHLA   RHLA   NO     CLOCK  /
22 065171  120  103  040  DH26:  .ASCII  /PC  TEST  PC OF  FAILING CONT.  CONT.  CONT.  CONT.  /<CRLF>
23 065272  040  040  040         .ASCIZ  /    NO    JSR    REG ADD RHCS1  RHCS2  RHDS1  RHER1  /
24 065373  120  103  040  DH27:  .ASCII  /PC  TEST  PC OF  FAILING GOOD   RECEIVED/<CRLF>
25 065454  040  040  040         .ASCIZ  /    NO    JSR    REG    DATA   DATA   /
26 065535  120  103  040  DH30:  .ASCII  /PC  TEST  PC OF  REG.   GOOD   BAD    /<CRLF>
27 065616  040  040  040         .ASCIZ  /    NO    JSR    ADDRESS DATA  DATA   /
28 065677  120  103  040  DH31:  .ASCII  /PC  TEST  WORD   GOOD   BAD    CONT.  CONT.  CONT   /<CRLF>
29 066000  040  040  040         .ASCIZ  /    NO    NO     DATA   DATA   RHCS1  RHDS1  RHER1  /
30 066101  120  103  040  DH32:  .ASCII  /PC  TEST  PC OF  WORD   GOOD   BAD    CONT.  CONT.  CONT.  /<C
31 066212  040  040  040         .ASCIZ  /    NO    JSR    NO     DATA   DATA   RHCS1  RHDS1  RHER1  /
32 066323  120  103  040  DH33:  .ASCII  /PC  TEST  PC OF  WORD   GOOD   CONT.  CONT.  CONT.  /<CRLF>
33 066424  040  040  040         .ASCIZ  /    NO    JSR    NO     DATA   RHCS1  RHDS1  RHER1  /
34 066525  120  103  040  DH34:  .ASCII  /PC  TEST  GOOD   GOOD   WRITTEN WRITTEN DATA   /<CRLF>
35 066616  040  040  040         .ASCIZ  /    NO    ECC1   ECC2   ECC1   ECC2   USED   /
36 066707  120  103  040  DH35:  .ASCII  /PC  TEST  GOOD   GOOD   PATTERN POSITON GOOD   RHER1  /<CRLF>
37 067010  040  040  040         .ASCIZ  /    NO    ECC1   ECC2   REG.REG.       POSITON REG.   /
38 067111  120  103  040  DH36:  .ASCII  /PC  TEST  PC OF  RHMR   POSITON PATTERN /<CRLF>
39 067172  040  040  040         .ASCIZ  /    NO    JSR    CONT.  REG.   REG./
40 067247  120  103  040  DH37:  .ASCII  /PC  TEST  POSITON POSITON GOOD   GOOD   PATTERN DATA   N-CODE/<CRL
41 067356  040  040  040         .ASCIZ  /    NO    ECC    GOOD   ECC1   ECC2   ECC    ENVLOPE ZEROS/
42
43 067464  120  103  040  DH40:  .ASCII  /PC  TEST  CONT./<CRLF>
44 067512  040  040  040         .ASCIZ  /    NO    RHWC/
45                              .EVEN
```

ERROR AND MESSAGE TABLE CONDIMENTS

```
 1 067540  001116  001432  041130  DT1:    .WORD   $ERRPC,TSTNM,REGADR,$GDDAT,$BDDAT,0
 2 067554  001116  001432  046426  DT2:    .WORD   $ERRPC,TSTNM,ERWORD,$GDDAT,0
 3 067566  001116  001432  046426  DT3:    .WORD   $ERRPC,TSTNM,ERWORD,$GDDAT,$BDDAT,0
 4
 5 067602  001116  001432  001122  DT11:   .WORD   $ERRPC,TSTNM,$BDADR,CS1,CS2,DS1,ER1,0
 6 067622  001116  001432  001122  DT14:   .WORD   $ERRPC,TSTNM,$BDADR,$BDDAT,CS1,CS2,DS1,ER1,0
 7 067644  001116  001432  001200  DT15:   .WORD   $ERRPC,TSTNM,$TMP1,0
 8 067654  001116  001432  001204  DT16:   .WORD   $ERRPC,TSTNM,$TMP3,$TMP1,$TMP0,$BDDAT,0
 9 067672  001116  001432  001310  DT17:   .WORD   $ERRPC,TSTNM,BA,DB,WC,CS1,CS2,0
10
11 067712  001116  001432  001316  DT20:   .WORD   $ERRPC,TSTNM,ER1,ER2,ER3,AS,DS1,0
12 067732  001116  001432  001314  DT21:   .WORD   $ERRPC,TSTNM,CS1,AS,DS1,0
13 067746  001116  001432  000000  DT22:   .WORD   $ERRPC,TSTNM,0
14 067754  001116  001432  001320  DT24:   .WORD   $ERRPC,TSTNM,DST,$BDDAT,$TMP1,$TMP2,$TMP3,0
15 067774  001116  001432  001414  DT26:   .WORD   $ERRPC,TSTNM,PCJSR,$BDADR,CS1,CS2,DS1,ER1,0
16 070016  001116  001432  001414  DT27:   .WORD   $ERRPC,TSTNM,PCJSR,REGADR,$GDDAT,$BDDAT,0
17
18 070034  001116  001432  001414  DT30:   .WORD   $ERRPC,TSTNM,PCJSR,REGADR,$GDDAT,$BDDAT,0
19 070052  001116  001432  046426  DT31:   .WORD   $ERRPC,TSTNM,ERWORD,$GDDAT,$BDDAT,CS1,DS1,ER1,0
20 070074  001116  001432  001414  DT32:   .WORD   $ERRPC,TSTNM,PCJSR,ERWORD,$GDDAT,$BDDAT,CS1,DS1,ER1,0
21 070120  001116  001432  001414  DT33:   .WORD   $ERRPC,TSTNM,PCJSR,ERWORD,$GDDAT,CS1,DS1,ER1,0
22 070142  001116  001432  044154  DT34:   .WORD   $ERRPC,TSTNM,GECC1,GECC2,WECC1,WECC2,DISK,0
23 070162  001116  001432  044154  DT35:   .WORD   $ERRPC,TSTNM,GECC1,GECC2,EC2,EC1,POSITI,ER1,0
24 070204  001116  001432  001414  DT36:   .WORD   $ERRPC,TSTNM,PCJSR,MR,EC1,EC2,0
25 070222  001116  001432  001344  DT37:   .WORD   $ERRPC,TSTNM,EC1,POSITI,GECC1,GECC2,EC2,DATENV,ZCODE,0
26
27 070246  001116  001432  001126  DT40:   .WORD   $ERRPC,TSTNM,$BDDAT,0
28
29 070256     000     000     000  DF1:    .BYTE   0,0,0,0,0
30 070263     000     000     001  DF2:    .BYTE   0,0,1,0
31 070267     000     000     001  DF3:    .BYTE   0,0,1,0,0
32
33 070274     000     000     000  DF11:   .BYTE   0,0,0,0,0,0,0
34 070303     000     000     000  DF14:   .BYTE   0,0,0,0,0,0,0,0
35 070313     000     000     000  DF15:   .BYTE   0,0,0
36 070316     000     000     000  DF16:   .BYTE   0,0,0,0,0
37 070323     000     000     000  DF17:   .BYTE   0,0,0,0,0,0,0
38
39 070332     000     000     000  DF20:   .BYTE   0,0,0,0,0,0,0
40 070341     000     000     000  DF21:   .BYTE   0,0,0,0,0
41 070346     000     000     000  DF22:   .BYTE   0,0,0,0
42 070352     000     000     000  DF24:   .BYTE   0,0,0,0,0,0,0
43 070361     000     000     000  DF26:   .BYTE   0,0,0,0,0,0,0,0
44 070371     000     000     000  DF27:   .BYTE   0,0,0,0,0,0
45
46 070377     000     000     000  DF30:   .BYTE   0,0,0,0,0,0
47 070405     000     000     001  DF31:   .BYTE   0,0,1,0,0,0,0,0
48 070415     000     000     000  DF32:   .BYTE   0,0,0,1,0,0,0,0,0
49 070426     000     000     000  DF33:   .BYTE   0,0,0,1,0,0,0,0
50 070436     000     000     000  DF34:   .BYTE   0,0,0,0,0,0,0
51 070445     000     000     000  DF35:   .BYTE   0,0,0,0,0,0,0,0
52 070455     000     000     000  DF36:   .BYTE   0,0,0,0,0,0
53 070463     000     000     000  DF37:   .BYTE   0,0,0,0,0,0,0,0,0
54
55 070474     000     000     000  DF40:   .BYTE   0,0,0
56                                        .EVEN
57
```

ERROR AND MESSAGE TABLE CONDIMENTS

```
     58          000200                      .END    200
```

SYMBOL TABLE

```
ABS    = 000200
ACL    = 000040
ACU    = 100000
ADTIMO  046034
AOE    = 001000
AS      001332
ATA    = 100000
ATABLE  003620
ATTENT  001416
AT0    = 000001
AT1    = 000002
AT2    = 000004
AT3    = 000010
AT4    = 000020
AT5    = 000040
AT6    = 000100
AT7    = 000200
A16    = 000400
A17    = 001000
BA      001310
BADTMO  004230
BAI    = 000010
BASECH  045104
BEGIN   004310
BEGIN2  004316
BITST   041132
BIT0   = 000001
BIT00  = 000001
BIT01  = 000002
BIT02  = 000004
BIT03  = 000010
BIT04  = 000020
BIT05  = 000040
BIT06  = 000100
BIT07  = 000200
BIT08  = 000400
BIT09  = 001000
BIT1   = 000002
BIT10  = 002000
BIT11  = 004000
BIT12  = 010000
BIT13  = 020000
BIT14  = 040000
BIT15  = 100000
BIT2   = 000004
BIT3   = 000010
BIT4   = 000020
BIT5   = 000040
BIT6   = 000100
BIT7   = 000200
BIT8   = 000400
BIT9   = 001000
BLNKS1  063203
BLNKS2  063202
BLNKS3  063201
BLNKS4  063200
BLNKS5  063177
BLNKS6  063176
BLNKS7  063175
BLNKS8  063174
BLT1    041160
BLT2    041202
BLT3    041212
BPTVEC= 000014
CA      001326
CAT     035652
CC      001352
CHECKD= 104414
CHECKE  041704
CHECKT  041514
CKSWR = 104407
CLAREA  041376
CLDISK  041460
CLR    = 000040
COMHD   046146
COMPA   047220
COMPAR  042354
COMWHD  051270
COUNTD  047666
CPHALT  062726
CPSAVE  056026
CR     = 000015
CRC     042666
CRLF   = 000200
CSF    = 000002
CSU    = 000010
CS1     001314
CS2     001312
CYL     046306
DAREAD  046362
DATENV  044172
DAWORD  046366
DB      001304
DCK    = 100000
DCLEAR  001450
DDISP  = 177570
DENVL  = 000200
DE1    = 000040
DFF20  = 000002
DF1     070256
DF11    070274
DF14    070303
DF15    070313
DF16    070316
DF17    070323
DF2     070263
DF20    070332
DF21    070341
DF22    070346
DF24    070352
DF26    070361
DF27    070371
DF3     070267
DF30    070377
DF31    070405
DF32    070415
DF33    070426
DF34    070436
DF35    070445
DF36    070455
DF37    070463
DF40    070474
DF5    = 000001
DH1     063205
DH11    063451
DH14    063630
DH15    064027
DH16    064060
DH17    064222
DH2     063327
DH20    064404
DH21    064566
DH22    064705
DH24    065007
DH26    065171
DH27    065373
DH30    065535
DH31    065677
DH32    066101
DH33    066323
DH34    066525
DH35    066707
DH36    067111
DH37    067247
DH40    067464
DIGB   = 000004
DISK    050224
DISPLA  001142
DISPRE  000174
DLT    = 100000
DL64   = 000020
DMD    = 000001
DOG     035206
DPR    = 000400
DRY    = 000200
DST     001320
DSWR   = 177570
DS1     001336
DT      001340
DTAGAP  051230
DTE    = 010000
DTSY   = 001000
DT1     067540
DT11    067602
DT14    067622
DT15    067644
DT16    067654
DT17    067672
DT2     067554
DT20    067712
DT21    067732
DT22    067746
DT24    067754
DT26    067774
DT27    070016
DT3     067566
DT30    070034
DT31    070052
DT32    070074
DT33    070120
DT34    070142
DT35    070162
DT36    070204
DT37    070222
DT40    070246
DVA    = 004000
ECDATA  044152
ECH    = 000100
ECI    = 004000
ECORR   044576
ECTEST  044210
EC1     001344
EC2     001346
EMTVEC= 000030
EM1     057072
EM11    057243
EM14    057276
EM15    057316
EM16    057366
EM17    057407
EM2     057155
EM20    057433
EM21    057454
EM22    057476
EM24    060147
EM25    060242
EM30    060302
EM31    060422
EM32    060544
EM33    061030
EM34    061102
EM35    061175
EM36    061471
EM37    061602
EM40    061736
EM6     057204
ERCLFC  017436
ERCRC   051464
ERCS2C  012302
ERFLG$  001406
ERHDGP  051466
ERHEAD  051462
ERPOS   044574
ERR    = 040000
ERROR  = 104000
ERRVEC= 000004
ERSTAR  046120
ERUNIT  046116
ERWORD  046426
ER1     001316
ER2     001322
ER3     001330
EXT1   = 000001
EXT10  = 000010
EXT2   = 000002
EXT20  = 000020
EXT4   = 000004
EXT40  = 000040
FEN    = 000200
FER    = 000020
FILLEC  044750
FMT22  = 010000
FNWORD  051504
FORMAT  047670
FOUT    051510
FRMAT1  032632
FSYNER  051460
FUTABL  001442
GCRC    051454
GECC1   044154
GECC2   044156
GO     = 000001
GRV    = 000010
GTSWR  = 104406
HADTMP  044176
HARDER  044170
HCCRCE  043270
HCE    = 000200
HCI    = 002000
HCRC   = 000400
HDESYN  051470
HDWSYN  050222
HEADER  050176
HEDGAP  046322
HEDSYN  046324
HEGAP   050210
HT     = 000011
IAE    = 002000
IBSAVE  056030
IE     = 000100
ILF    = 000001
ILLEGL  001504
ILR    = 000002
IOTVEC= 000020
IR     = 000100
IXE    = 004000
JP1     036164
JP2     036200
KEY1    046312
KEY2    046314
KIPAR0= 172340
KIPAR1= 172342
KIPAR2= 172344
KIPAR3= 172346
KIPAR4= 172350
```

SYMBOL TABLE

```
KIPAR5= 172352        OR     = 000200        PRER1   010166        RHEC2   001272        SR2    = 177576
KIPAR6= 172354        OUT      046372        PRER2   010464        RHER1   001242        SR3    = 172516
KIPAR7= 172356        PAR    = 000010        PRER3   010640        RHER2   001246        SSECTR  051530
KIPDR0= 172300        PAT    = 000020        PROF    010530        RHER3   001254        SSYN    046320
KIPDR1= 172302        PCJSR    001414        PROG   = 001000        RHLA    001274        STACK  = 001100
KIPDR2= 172304        PFECH    056230        PRWC    010056        RHMR    001260        START   004324
KIPDR3= 172306        PFECH1   056240        PR0    = 000000        RHOF    001250        STKLMT= 177774
KIPDR4= 172310        PFECH2   056322        PR1    = 000040        RHSN    001266        SWR     001140
KIPDR5= 172312        PFECH3   056354        PR2    = 000100        RHWC    001232        SWREG   000176
KIPDR6= 172314        PFECH4   056364        PR3    = 000140        RH70    001436        SW0    = 000001
KIPDR7= 172316        PFTSTN   056370        PR4    = 000200        RH70CK  004736        SW00   = 000001
LA      001350        PGE    = 002000        PR5    = 000240        RKEY1   046416        SW01   = 000002
LAD     041440        PIE1   = 100000        PR6    = 000300        RKEY2   046420        SW02   = 000004
LERR    041126        PIE10  = 000100        PR7    = 000340        RMR    = 000004        SW03   = 000010
LF     = 000012       PIE11  = 000040        PS     = 177776        RNCTR1  051266        SW04   = 000020
LST    = 002000       PIE12  = 000020        PSEL   = 002000        RNO     052624        SW05   = 000040
MAKECY  044032        PIE13  = 000010        PSU    = 000001        RNWAT1  051346        SW06   = 000100
MASK    041124        PIE14  = 000004        PSW    = 177776        RPVEC   001226        SW07   = 000200
MCLK   = 000002       PIE15  = 000002        PUTREG  041064        RPVECT  052740        SW08   = 000400
MCPE   = 020000       PIE16  = 000001        PWRVEC= 000024        RP06    001434        SW09   = 001000
MHS    = 001000       PIE17  = 100000        P12     044202        RSETR   046414        SW1    = 000002
MID     044022        PIE18  = 040000        P22     044204        RSYNC   046410        SW10   = 002000
MIDDLE  043720        PIE19  = 020000        P24     044206        RUNCTR  046144        SW11   = 004000
MINX   = 000004       PIE2   = 040000        P3      044200        RUNWAT  046224        SW12   = 010000
MMVEC  = 000250       PIE20  = 010000        RAW    = 000020        R6     =%000006       SW13   = 020000
MOL    = 010000       PIE21  = 004000        RCOM    052626        R7     =%000007       SW14   = 040000
MPE    = 000400       PIE22  = 002000        RCYL    046412        SAVDT   001410        SW15   = 100000
MR      001334        PIE23  = 001000        RDCHR  = 104410        SAVER   042152        SW2    = 000004
MRD    = 000020       PIE24  = 000400        RDHEAD  046430        SAVSN   001412        SW3    = 000010
MSE    = 000020       PIE25  = 000200        RDLIN  = 104411        SC     = 100000        SW4    = 000020
MSTCK  = 000010       PIE26  = 000100        RDOCT  = 104412        SCOPE  = 000004        SW5    = 000040
MWR    = 000040       PIE27  = 000040        RDY    = 000200        SCOP1  = 104413        SW6    = 000100
MXF    = 001000       PIE28  = 000020        READ    047222        SCRC    051536        SW7    = 000200
NCODE   044162        PIE29  = 000010        READAT  001466        SCYL    051526        SW8    = 000400
NCOUNT  044164        PIE3   = 020000        READIN  001502        SC1    = 000100        SW9    = 001000
NDRVAS  062426        PIE30  = 000004        RECALI  001446        SC10   = 001000        TAGDTE  001430
NED    = 010000       PIE31  = 000002        REDATA  052630        SC2    = 000200        TBITVE= 000014
NEM    = 004000       PIE32  = 000001        REFOR   001470        SC20   = 002000        TDF    = 000040
NHS    = 002000       PIE4   = 010000        REGADR  041130        SC4    = 000400        TESDTE  001426
NOPERA  001442        PIE5   = 004000        REGSA1  055504        SEARCH  052354        TESTAD  040306
NOSYNC  046422        PIE6   = 002000        REINTO  002554        SECGAP  050126        TIMCNT  042030
NOUNIT  001376        PIE7   = 001000        RELEAS  001452        SECOTR  046310        TKVEC  = 000060
NOWORD  046354        PIE8   = 000400        RESVEC= 000010        SECTR   052352        TMPILL  001422
NUNIT   001400        PIE9   = 000200        RETCL   001476        SEECOM  001472        TOLGAP  051232
OCYL   = 100000       PIP    = 020000        RHAS    001256        SEGPER  051456        TOTALA  001420
OF      001324        PIRQ   = 177772        RHBA    001234        SELECT  001402        TPVEC  = 000064
OFREV  = 000200       PIRQVE= 000240         RHBAE   001300        SELTST  006424        TRAPVE= 000034
OFSETC  001474        PKACK    001500        RHCA    001252        SERCH   001454        TRE    = 040000
OF100  = 000004       PLU    = 020000        RHCC    001276        SETDSK  043142        TRK     046260
OF200  = 000010       POSITI   044166        RHCS1   001240        SILOSZ  001440        TRK1   = 004000
OF25   = 000001       PRBA     010122        RHCS2   001236        SILOTB  050224        TRK10  = 040000
OF400  = 000020       PRCA     010574        RHCS3   001302        SKEY1   051532        TRK2   = 010000
OF50   = 000002       PRCS1    007642        RHDB    001230        SKEY2   051534        TRK20  = 100000
OF800  = 000040       PRCS2    007576        RHDST   001244        SKI    = 040000        TRK4   = 020000
OPDCL  = 000100       PRDST    010420        RHDS1   001262        SN      001342        TRTVEC= 000014
OPERSE  040310        PRE    = 000020        RHDT    001264        SR0    = 177572        TSECC   001424
OPI    = 020000       PREAMB   062040        RHEC1   001270        SR1    = 177574        TSECCG  044160
```

SYMBOL TABLE

```
TSTNM   001432
TST1    005052
TST10   007620
TST100  033444
TST101  033500
TST102  033622
TST103  033656
TST104  034000
TST105  034034
TST106  034134
TST107  034170
TST11   007664
TST110  034270
TST111  034416
TST112  034536
TST113  034572
TST114  035206
TST115  035242
TST116  035652
TST117  036476
TST12   010034
TST120  036532
TST121  037064
TST122  037410
TST123  037740
TST13   010100
TST14   010144
TST15   010210
TST16   010376
TST17   010442
TST2    005534
TST20   010506
TST21   010552
TST22   010616
TST23   010662
TST24   011270
TST25   012314
TST26   012456
TST27   013026
TST3    005620
TST30   013204
TST31   013432
TST32   013672
TST33   014016
TST34   014316
TST35   014454
TST36   014574
TST37   014752
TST4    005672
TST40   015076
TST41   015222
TST42   015516
TST43   015742
TST44   016332

TST45   017464
TST46   020402
TST47   021056
TST5    006440
TST50   021464
TST51   022030
TST52   022374
TST53   022552
TST54   022606
TST55   023474
TST56   023530
TST57   024040
TST6    007202
TST60   024376
TST61   024736
TST62   025202
TST63   025566
TST64   026152
TST65   026536
TST66   027076
TST67   027442
TST7    007530
TST70   027772
TST71   030300
TST72   030670
TST73   031234
TST74   031600
TST75   032206
TST76   032610
TST77   033140
TUF     =  000100
TY         046424
TYPDS   =  104405
TYPE    =  104401
TYPOC   =  104402
TYPON   =  104404
TYPOS   =  104403
T.SCOP     041442
UN         005610
UNIT       001374
UNITS      001354
UNITSL     001404
UNLOAD     001444
UNS     =  040000
UPE     =  020000
US1     =  000001
US2     =  000002
US4     =  000004
UWR     =  000010
VAR1       036146
VAR2       036154
VUF     =  000002
VU30    =  010000

VV      =  000100
WAIT.T     042032
WAT     =  104415
WC         001306
WCE     =  040000
WCF     =  000040
WCRC       050206
WCU     =  000001
WCYL       051444
WECC1      051224
WECC2      051226
WKEY1      051450
WKEY2      051452
WLE     =  004000
WORD       047216
WRCHDA     042516
WRCHDT     001460
WRCHEK     001456
WRCHHD     042204
WRDATA     047674
WRFROM     001510
WRHEAD     051540
WRIDAT     001462
WRIFOR     001464
WRITE      047454
WRL     =  004000
WRU     =  000400
WSECTR     051446
WSSYNC     050174
WSU     =  000004
WTRK       051402
WWORD      047452
X          046316
XE2        006076
Y          046356
ZCODE      044174
ZER     =  000400
ZWORDS     047672
$AUTOB     001134
$BDADR     001122
$BDDAT     001126
$BELL      001216
$CHARC     054172
$CKSWR     054516
$CMTAG     001100
$CM1    =  000006
$CM2    =  000014
$CM3    =  000006
$CM4    =  000006
$CNTLC     055267
$CNTLG     055301
$CNTLU     055274
$CRLF      001223

$DBLK      053674
$DOAGN     040262
$DTBL      053664
$ENDAD     040252
$ENDCT     040220
$ENDMG     040271
$ENULL     040266
$EOP       040164
$EOPCT     040212
$ERFLG     001103
$ERMAX     001115
$ERROR     055470
$ERRPC     001116
$ERRTB     003630
$ERRTY     056032
$ERTTL     001112
$ESCAP     001214
$FILLC     001156
$FILLS     001155
$GDADR     001120
$GDDAT     001124
$GET42     040242
$GTSWR     054606
$HD     =  000000
$HIOCT     055466
$ICNT      001104
$ILLUP     057054
$INTAG     001135
$ITEMB     001114
$LF        001224
$LPADR     001106
$LPERR     001110
$MNEW      055317
$MSWR      055306
$MXCNT     053456
$NULL      001154
$NWTST=    000001
$OCNT      056614
$OMODE     056616
$OVER      053442
$PASS      001100
$POWER     057062
$PWRAD     057050
$PWRDN     056710
$PWRMG     057044
$PWRUP     056762
$QUES      001222
$RDCHR     055060
$RDLIN     055150
$RDOCT     055330
$RDSZ   =  000011
$REGAD     001160
$REGO      001162

$REG1      001164
$REG2      001166
$REG3      001170
$REG4      001172
$REG5      001174
$RTNAD     040264
$SAVR6     057060
$SCOPE     053020
$SETUP=    000117
$SS1    =  000000
$STUP   =  177777
$SVLAD     053414
$SVPC   =  000214
$SWR    =  167770
$SWRMK=    000000
$TIMES     001212
$TKB       001146
$TKCNT     054176
$TKINT     054216
$TKQEN=    054215
$TKQIN     054200
$TKQOU     054202
$TKQSR     054204
$TKS       001144
$TKSRV     054266
$TMP0      001176
$TMP1      001200
$TMP2      001202
$TMP3      001204
$TMP4      001206
$TMP5      001210
$TN     =  000124
$TPB       001152
$TPFLG     001157
$TPS       001150
$TRAP      056620
$TRAP2     056642
$TRP    =  000016
$TRPAD     056654
$TSTNM     001102
$TTYIN     055256
$TYPDS     053460
$TYPE      053704
$TYPEC     054054
$TYPEX     054174
$TYPOC     056416
$TYPON     056432
$TYPOS     056372
$XOFF   =  000023
$XON    =  000021
$XTSTR     053056
$$GET4=    000000
$OFILL     056615

. ABS.  070500  000
        000000  001
```

SYMBOL TABLE

ERRORS DETECTED: 0

VIRTUAL MEMORY USED: 57856 WORDS (226 PAGES)
DYNAMIC MEMORY AVAILABLE FOR 71 PAGES
CZRJGE.BIC,CZRJGE=CZRJGE.DOC,CZRJGE,[20,0]SYSMAC/M