RP04/5/6

DISKLESS CONTROLLER PART 1

MD-11-DZRJG-A

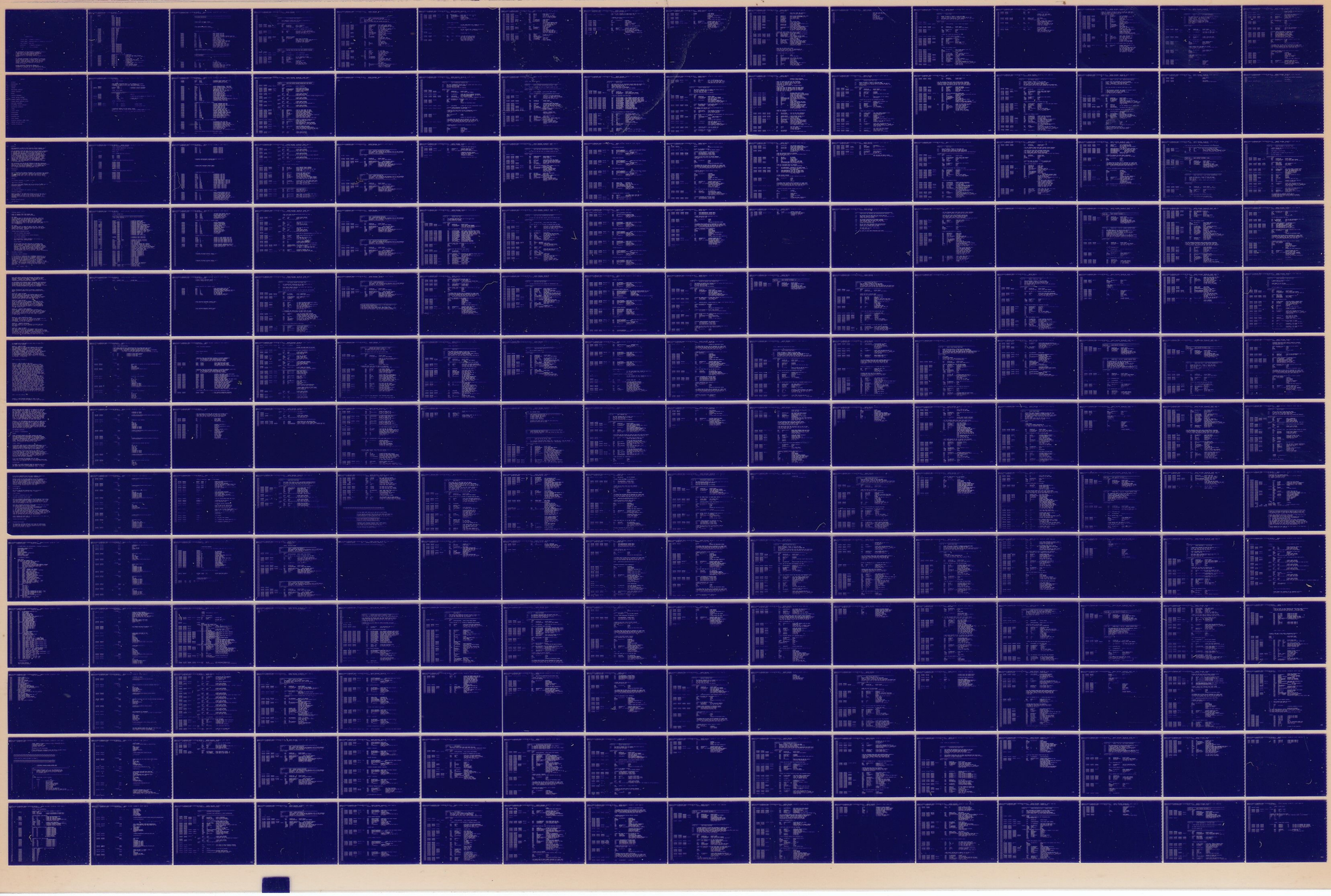
EP-DZRJG-A-DL-A

NOV 1976

digital

FICHE 1 OF 2

MADE IN USA



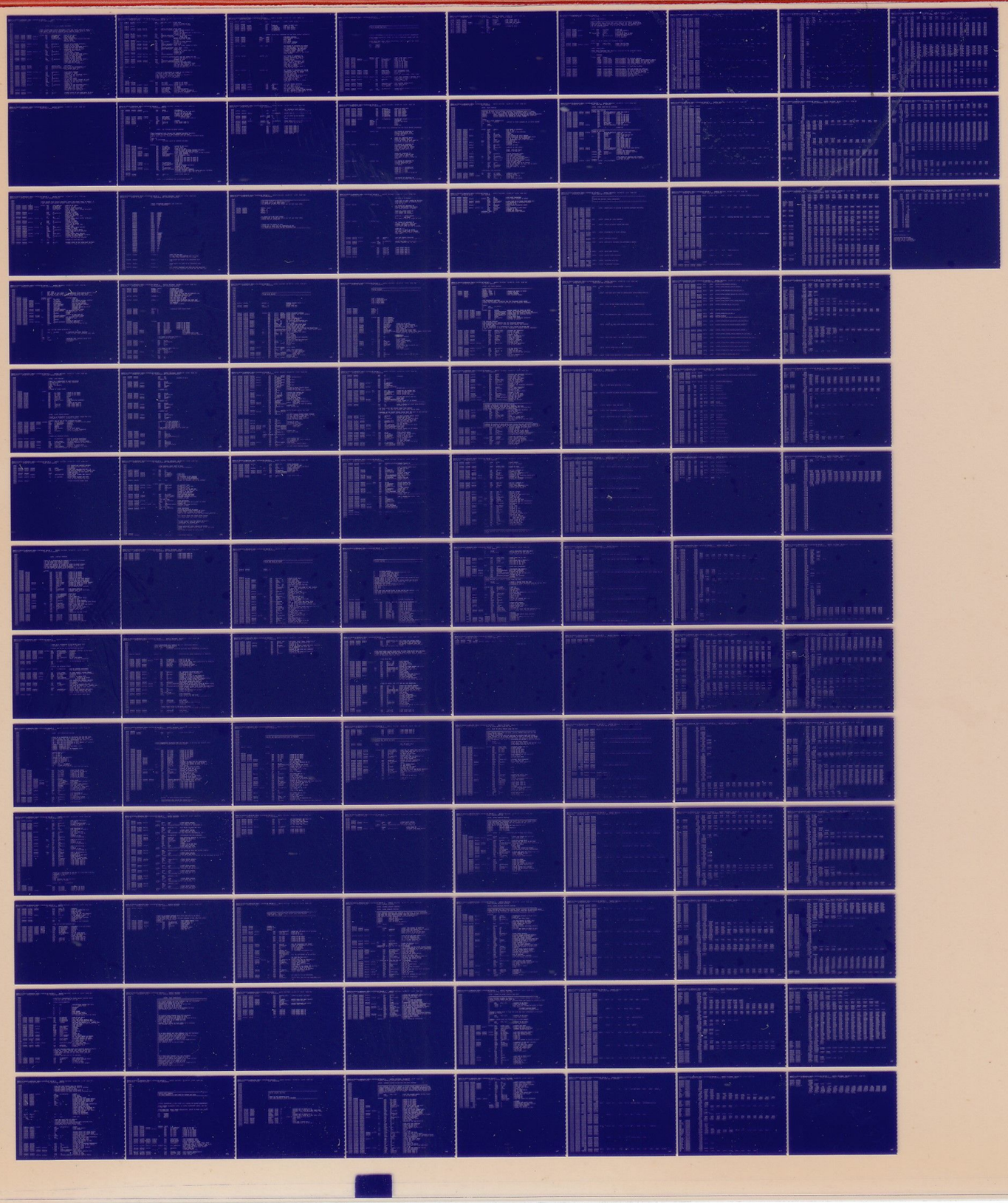
RP04/5/6

DISKLESS CONTROLLER PART 1
MD-11-DZRJG-A

EP-DZRJG-A-DL-A

FICHE 2 OF 2

NOV 1976
digital
MADE IN USA



IDENTIFICATION

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

PRODUCT NAME: RP04/5/6 DISKLESS CONTROLLER TEST-PART I

DATE CREATED: MAY 1976

MAINTAINER: DIAGNOSTIC ENGINEERING

AUTHOR: PETE BLACKSTONE

A1

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

C1

4.2 STARTING ADDRESS

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

200 START
ALL SWITCHES MUST BE DOWN FOR WORST CASE RUN. WITH THIS STARTING ADDRESS ALL THE 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.

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.

8.4 PROGRAM REVISION HISTORY

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.

```
                        .TITLE  MNDEC-11-DZRJG-A,RP04/5/6 DSKLS CONTROLLER TST-PT 1
                        ;*COPYRIGHT (C) 1976
                        ;*DIGITAL EQUIPMENT CORP.
                        ;*MAYNARD, MASS. 01754
                        ;*
                        ;*PROGRAM BY PETE BLACKSTONE
                        ;*
                        ;*THIS PROGRAM WAS ASSEMBLED USING THE PDP-11 MAINDEC SYSMAC
                        ;*PACKAGE (MAINDEC-11-DZQAC-C3), JAN 19, 1977.
                        ;*

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

;*DRIVE MUST BE LOCKED ON PORT A OR PORT B

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


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

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


                        .SBTTL  OPERATIONAL SWITCH SETTINGS
                        ;*
                        ;*      SWITCH                  USE
                        ;*      ------          --------------------
                        ;*        15            HALT ON ERROR
                        ;*        14            LOOP ON TEST
                        ;*        13            INHIBIT ERROR TYPEOUTS
                        ;*        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
```

L1

```
                        .SBTTL  BASIC DEFINITIONS

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

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

                        ;*GENERAL PURPOSE REGISTER DEFINITIONS
000000                  R0=     %0              ;;GENERAL REGISTER
000001                  R1=     %1              ;;GENERAL REGISTER
000002                  R2=     %2              ;;GENERAL REGISTER
000003                  R3=     %3              ;;GENERAL REGISTER
000004                  R4=     %4              ;;GENERAL REGISTER
000005                  R5=     %5              ;;GENERAL REGISTER
000006                  R6=     %6              ;;GENERAL REGISTER
000007                  R7=     %7              ;;GENERAL REGISTER
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
```

```
                        SW01=   2
                        SW00=   1
                        .EQUIV  SW09,SW9
                        .EQUIV  SW08,SW8
                        .EQUIV  SW07,SW7
                        .EQUIV  SW06,SW6
                        .EQUIV  SW05,SW5
                        .EQUIV  SW04,SW4
                        .EQUIV  SW03,SW3
                        .EQUIV  SW02,SW2
                        .EQUIV  SW01,SW1
                        .EQUIV  SW00,SW0

                        ;*DATA BIT DEFINITIONS (BIT00 TO BIT15)
100000                  BIT15=  100000
040000                  BIT14=  40000
020000                  BIT13=  20000
010000                  BIT12=  10000
004000                  BIT11=  4000
002000                  BIT10=  2000
001000                  BIT09=  1000
000400                  BIT08=  400
000200                  BIT07=  200
000100                  BIT06=  100
000040                  BIT05=  40
000020                  BIT04=  20
000010                  BIT03=  10
000004                  BIT02=  4
000002                  BIT01=  2
000001                  BIT00=  1
                        .EQUIV  BIT09,BIT9
                        .EQUIV  BIT08,BIT8
                        .EQUIV  BIT07,BIT7
                        .EQUIV  BIT06,BIT6
                        .EQUIV  BIT05,BIT5
                        .EQUIV  BIT04,BIT4
                        .EQUIV  BIT03,BIT3
                        .EQUIV  BIT02,BIT2
                        .EQUIV  BIT01,BIT1
                        .EQUIV  BIT00,BIT0

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

```
                                .SBTTL  TRAP CATCHER

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

                                .SBTTL  ACT11 HOOKS

                                ;;*****************************************************************
                                ;HOOKS REQUIRED BY ACT11
                        000200          $SVPC=.                 ;SAVE PC
                        000046          .=46
000046  041322                          $ENDAD                  ;;1)SET LOC.46 TO ADDRESS OF $ENDAD IN .$EOP
                        000052          .=52
000052  020000                          .WORD   20000           ;;2)SET LOC.52 TO 20000
                        000200          .=$SVPC                 ;; RESTORE PC

                                .SBTTL  STARTING ADDRESS

                        000200          .=200
000200  000137  004240                  JMP     @#BEGIN                 ;NORMAL START
                        000210          .=210
000210  000137  004230                  JMP     @#BEGIN2                ;SELECT DRIVE START


                                ;**STARTING ADDRESS 200 FOR NORMAL STARTS
                                ;**THIS WILL TEST ALL DRIVES ON THE SYSTEM A SINGLE DRIVE AT A TIME
                                ;**
                                ;**STARTING ADDRESS 210 WILL TEST ONLY ONE SPECIFIED DRIVE

```

```
                        .SBTTL  MEMORY MANAGEMENT DEFINITIONS

                        ;*KT11 VECTOR ADDRESS

000250                  MMVEC=  250

                        ;*KT11 STATUS REGISTER ADDRESSES

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

                        ;*KERNEL "I" PAGE DESCRIPTOR REGISTERS

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

                        ;*KERNEL "I" PAGE ADDRESS REGISTERS

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

                        ;;**************************************************************************
001110                          .=1110                          ; ?
```

C2

```
                                .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          $REGAD: .WOPD   0               ;;CONTAINS THE ADDRESS FROM
                                                        ;;WHICH  ($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:  .WOPD   0               ;;USER DEFINED
001210  000000          $TMP5:  .WORD   0               ;;USER DEFINED
001212  000000          $TIMES: 0                       ;;MAX. NUMBER OF ITERATIONS
001214  000000          $ESCAPE:0                       ;;ESCAPE ON ERROR ADDRESS
001216  177607  000377    $BELL:  .ASCIZ  <207><377><377> ;;CODE FOR BELL
001222  077             $QUES:  .ASCII  /?/             ;;QUESTION MARK
001223  015             $CRLF:  .ASCII  <15>            ;;CARRIAGE RETURN
```

D2

```
   290  001224  000012                  $LF:    .ASCIZ  <12>            ;;LINE FEED
   291                                  ;;*************************************************************************
```

```
                                .SBTTL  ERROR POINTER TABLE

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

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


001226                          $ERRTB:



                                ;*ITEM1
001226  057154                          EM1             ;WRONG DATA IN READING OR WRITING HARDWARE REGISTER
001230  062430                          DH1             ;PC
                                                        ;REG. ADDR.
                                                        ;GOOD DATA
                                                        ;RECEIVED DATA
001232  067010                          DT1             ;$ERRPC,REGADR,$GDDAT,$BDDAT
001234  067526                          DF1             ;0,0,0,0,0



                                ;*ITEM2
001236  057237                          EM2             ;ERROR ON  DATA COMMAND

001240  065565                          DH33            ;PC
                                                        ;PC OF JSR
                                                        ;TEST NO
                                                        ;WORD NO.
                                                        ;GOOD DATA
                                                        ;CONTENTS OF RHCS1
                                                        ;CONTENTS OF RHDS1
                                                        ;CONTENTS OF RHER1
001242  067370                          DT33            ;$ERRPC,PCJSR,$TSTNM,ERWORD,$GDDAT,CS1,DS1,ER1
001244  067676                          DF33            ;0,0,0,1,0,0,0,0



                                ;*ITEM3
001246  057237                          EM2             ;ERROR ON  DATA COMMAND

001250  065342                          DH32            ;PC
                                                        ;PC OF JSR
                                                        ;TEST NO
                                                        ;WORD NO.
                                                        ;GOOD DATA
                                                        ;BAD DATA
                                                        ;CONTENTS OF RHCS1
```

```
                                                        ;CONTENTS OF RHDS1
                                                        ;CONTENTS OF RHER1

001252  067344                  DT32                  ;$ERRPC,PCJSR,$TSTNM,ERWORD,$GDDAT,$BDDAT,CS1,DS1,ER1
001254  067665                  DF32                  ;0,0,0,1,0,0,0,0,0,

                        ;*ITEM4
001256  057237                  EM2                   ;ERROR ON  DATA COMMAND

001260  065137                  DH31                  ;PC
                                                      ;TEST NO
                                                      ;WORD NO.
                                                      ;GOOD DATA
                                                      ;BAD DATA
                                                      ;CONTENTS OF RHCS1
                                                      ;CONTENTS OF RHDS1
                                                      ;CONTENTS OF RHER1

001262  067322                  DT31                  ;$ERRPC,$TSTNM,ERWORD,$GDDAT,$BDDAT,CS1,DS1,ER1
001264  067655                  DF31                  ;0,0,1,0,0,0,0,0,

                        ;*ITEM5
001266  000000                  0                     ;
001270  000000                  0                     ;
001272  067322                  DT31                  ;$ERRPC,$TSTNM,ERWORD,$GDDAT,$BDDAT,CS1,DS1,ER1
001274  067655                  DF31                  ;0,0,1,0,0,0,0,0,

                        ;*ITEM6
001276  057266                  EM6                   ;ERROR ON WRITE HEADER AND DATA

001300  065342                  DH32                  ;PC
                                                      ;PC OF JSR
                                                      ;TEST NO
                                                      ;WORD NO.
                                                      ;GOOD DATA
                                                      ;BAD DATA
                                                      ;CONTENTS OF RHCS1
                                                      ;CONTENTS OF RHDS1
                                                      ;CONTENTS OF RHER1

001302  067344                  DT32                  ;SERRPC,PCJSR,$TSTNM,ERWORD,$GDDAT,$BDDAT,CS1,DS1,ER1
001304  067665                  DF32                  ;0,0,0,1,0,0,0,0,0,

                        ;*ITEM7
001306  057266                  EM6                   ;ERROR ON WRITE HEADER AND DATA
001310  062553                  DH2                   ;PC
                                                      ;TEST NO
                                                      ;WORD NO.
                                                      ;GOOD DATA
                                                      ;BAD DATA
```

G2

```
404  001312  067036                    DT3                     ;$ERRPC,$TSTNM,ERWORD,$GDDAT,$BDDAT
405  001314  067537                    DF3                     ;0,0,1,0,0,
406
407
408                                ;*ITEM10
409  001316  000000                    0                       ;
410  001320  000000                    0                       ;
411  001322  067036                    DT3                     ;$ERRPC,$TSTNM,ERWORD,$GDDAT,$BDDAT
412  001324  067537                    DF3                     ;0,0,1,0,0,
413
414
415                                ;*ITEM11
416  001326  057325                    EM11                    ;CONTROLLER OR DRIVE STATUS
417  001330  062676                    DH11                    ;PC
418                                                            ;TEST NO
419                                                            ;FAILING REG. ADDR
420                                                            ;CONTENTS OF RHCS1
421                                                            ;CONTENTS OF RHCS2
422                                                            ;CONTENTS OF RHDS1
423                                                            ;CONTENTS OF RHER1
424  001332  067052                    DT11                    ;$ERRPC,$TSTNM,$BDADR,CS1,CS2,DS1,ER1
425  001334  067544                    DF11                    ;0,0,0,0,0,0
426
427
428                                ;*ITEM12
429  001336  057325                    EM11                    ;WRONG DATA FROM SILO
430
431  001340  062430                    DH1                     ;PC
432                                                            ;REG.ADDR
433                                                            ;GOOD DATA
434                                                            ;RECEIVED DATA
435  001342  067010                    DT1                     ;$ERRPC,REGADR,$GDDAT,$BDDAT
436  001344  067526                    DF1                     ;0,0,0,0
437
438
439                                ;*ITEM13
440  001346  000000                    0
441  001350  000000                    0
442  001352  067010                    DT1                     ;$ERRPC,TSTNM,REGADR,$GDDAT,$BDDAT
443  001354  067526                    DF1                     ;0,0,0,0,0
444
445
446                                ;*ITEM14
447  001356  057360                    EM14                    ;REGISTER FAILED
448  001360  063056                    DH14                    ;PC
449                                                            ;FAILING REG. ADDP
450                                                            ;CONTENTS OF FAILING REG.
451                                                            ;CONTENTS OF RHCS1
452                                                            ;CONTENTS OF RHCS2
453                                                            ;CONTENTS OF RHDS1
454                                                            ;CONTENTS OF RHER1
455  001362  067072                    DT14                    ;$ERRPC,$BDADR,$BDDAT,CS1,CS2,DS1,ER1
456  001364  067553                    DF14                    ;0,0,0,0,0,0,0
457
458
459                                ;*ITEM15
```

H2

```
460  001366  057400                    EM15                ;SPECIFIED REG. NON EXISTANT SO ABORT
461                                                          ;PROGRAM
462  001370  063256                    DH15                ;PC
463                                                          ;ADDR. OF REG
464  001372  067114                    DT15                ;$ERRPC,TEMP1
465  001374  067563                    DF15                ;0,0
466
467
468                                ;*ITEM16
469  001376  057451                    EM16                ;WAIT LOOP FAILED
470  001400  063307                    DH16                ;PC
471                                                          ;WAT PC
472                                                          ;BIT WANTED
473                                                          ;REG. ADR.
474                                                          ;REG. CONT.
475  001402  067124                    DT16                ;$ERRPC,$TMP3,$TMP1,$TMP0,$BDDAT
476  001404  067566                    DF16                ;0,0,0,0
477
478
479                                ;*ITEM17
480  001406  057472                    EM17                ;WRITE CHECK FAILING
481  001410  063452                    DH17                ;PC
482                                                          ;TEST NO
483                                                          ;CONTENTS OF RHBA
484                                                          ;CONTENTS OF RHDB
485                                                          ;CONTENTS OF RHWC
486                                                          ;CONTENTS OF RHCS1
487                                                          ;CONTENTS OF RHCS2
488  001412  067142                    DT17                ;$ERRPC,$TSTNM,$BA,DB,WC,CS1,CS2
489  001414  067573                    DF17                ;0,0,0,0,0,0,0
490
491
492                                ;*ITEM20
493  001416  057516                    EM20                ;REGISTER FAILING
494  001420  063635                    DH20                ;PC
495                                                          ;TST NO
496                                                          ;CONTENTS OF RHER1
497                                                          ;CONTENTS OF RHER2
498                                                          ;CONTENTS OF RHER3
499                                                          ;CONTENTS OF RHAS
500                                                          ;CONTENTS OF RHDS1
501  001422  067162                    DT20                ;$ERRPC,TSTNM ER1,ER2,ER3,AS,DS1
502  001424  067602                    DF20                ;0,0,0,0,0,0,0
503
504                                ;*ITEM21
505
506  001426  057537                    EM21                ;INTERRUPT FAILING
507  001430  064020                    DH21                ;PC
508                                                          ;TEST NO
509                                                          ;CONTENTS OF RHCS1
510                                                          ;CONTENTS OF RHAS
511                                                          ;CONTENTS OF RHDS1
512  001432  067202                    DT21                ;$ERRPC,TSTNM,CS1,AS,DS1
513  001434  067611                    DF21                ;0,0,0,0,0
514
515
```

```
                        ;*ITEM22
001436  057561                EM22            ;ERROR IN DRIVE PRESENT -
                                                ;LOOKING AT RHAS AND RHCS2-NED(BIT#12)
                                                ;DRIVES PRESENT DO NOT AGREE
                                                ;NOTE: ON DUAL PORT SYSTEM
                                                ;DRIVE ON OTHER PORT WILL NOT GIVE NED
                                                ;HENCE THERE WILL BE A MISSMATCH
001440  064140                DH22            ;PC
                                                ;TEST NO
                                                ;RHAS UNIT (RHER1 BITS SET)
                                                ;RHCS2 UNIT ('NED' BIT TEST)
001442  067216                DT22            ;$ERRPC,TSTNM
001444  067616                DF22            ;0,0

                        ;*ITEM23
001446  000000                0               ;NO LONGER USED DUE TO SPECIAL 'NED'
001450  000000                0               ;TEST TABLE TYPE OUT ROUTINE
001452  000000                0
001454  000000                0

                        ;*ITEM 24
001456  060241                EM24            ;LOOK AHEAD REGISTER AT THE
                                                ;BEGINNING OF A SECTOR IS IN
                                                ;ERROR
001460  064243                DH24            ;PC
                                                ;RHDST
                                                ;BAD RHLA
                                                ;GOOD RHLA
                                                ;SECTOR NO
                                                ;SECTOR CLOCK
001462  067224                DT24            ;$ERRPC,DST,$BDDAT,$TMP1,$TMP2,$TMP3
001464  067622                DF24            ;0,0,0,0,0

                    ;*ITEM 25
001466  060334                EM25            ;LOOK AHEAD REGISTER IS
                                                ;IN ERROR
001470  064243                DH24            ;PC
                                                ;RHDST
                                                ;BAD RHLA
                                                ;GOOD RHLA
                                                ;SECTOR NO
                                                ;SECTOR CLOCK
001472  067224                DT24            ;$ERRPC,DST,$BDDAT,$TMP1,$TMP2,$TMP3
001474  067622                DF24            ;0,0,0,0,0
                        ;*ITEM26
001476  057325                EM11            ;CONTROLLER OR DRIVE STATUS

001500  064426                DH26            ;PC
                                                ;PC OF JSR
                                                ;FAILING REGISTER ADDRESS
                                                ;CONTENTS OF RHCS1
                                                ;CONTENTS OF RHCS2
                                                ;CONTENTS OF RHDS1
```

```
                                                        ;CONTENTS OF RHER1

001502  067244                  DT26                  ;$ERRPC,PCJSR,$BDADR,CS1,CS2,DS1,ER1
001504  067631                  DF26                  ;0,0,0,0,0,0,


                        ;*ITEM27
001506  057154                  EM1                   ;ERROR IN READING OR WRITING HARDWARE REGISTER

001510  064631                  DH27                  ;PC
                                                      ;PC OF JSR
                                                      ;TEST NUMBER
                                                      ;FAILING REGISTER
                                                      ;GOOD DATA
                                                      ;RECEIVED DATA

001512  067266                  DT27                  ;$ERRPC,PCJSR,TSTNM,REGADR,$GDDAT,$BDDAT
001514  067641                  DF27                  ;0,0,0,0,0,0


                        ;*ITEM30
001516  060374                  EM30                  ;CURRENT CYLINDER DOES NOT REFLECT DESIRED CYLINDER REG.
001520  064774                  DH30                  ;PC
                                                      ;PC OF JSR
                                                      ;REGISTER ADDRESS
                                                      ;GOOD DATA
                                                      ;BAD DATA

001522  067304                  DT30                  ;$ERRPC,PCJSR,REGADR,$GDDAT,$BDDAT
001524  067647                  DF30                  ;0,0,0,0,0


                      ;*ITEM31
001526  060516                  EM31                  ;ECC GENERATED IS INCORRECT
                                                      ;EVERY WORD IN THIS SECTOR IS GIVEN IN "DATA USED"

001530  065770                  DH34                  ;PC
                                                      ;TEST NUMBER
                                                      ;GOOD ECC1
                                                      ;GOOD EC2C
                                                      ;WRITTEN ECC1
                                                      ;WRITTEN ECC2
                                                      ;DATA USED

001532  067412                  DT34                  ;$ERRPC,TSTNM,GECC1,GECC2,WECC1,WECC2,DISK

001534  067706                  DF34                  ;0,0,0,0,0,0,0


                      ;*ITEM32
001536  060641                  EM32                  ;ON READ COMMAND AFTER DATA AND ECC HAVE BEEN READ
                                                      ;ECC REGISTER OR RHER1 IS IN ERROR
                                                      ;ONLY LOWER 11 BITS OF PATTERN REGISTER
```

```
                                                        ;CAN BE READ
                                                        ;THIS SHUOLD MATCH LOWER 11 BITS OF ECC1
001540  066153                      DH35                ;PC
                                                        ;TEST NUMBER
                                                        ;GOOD ECC1
                                                        ;GOOD ECC2
                                                        ;PATTERN REGISTER
                                                        ;RHER1
001542  067432                      DT35                ;$ERRPC,TSTNM,GECC1,GECC2,EC2,ER1
001544  067715                      DF35                ;0,0,0,0,0,0


                            ;*ITEM33
001546  061130                      EM33                ;HIGH COUNT BIT NOT HIGH AFTER 38859 CLOCKS
001550  066356                      DH36                ;PC
                                                        ;PC OF JSR
                                                        ;TEST NUMBER
                                                        ;RHMR
                                                        ;POSITION REG.
                                                        ;PATTERN REGISTER
001552  067454                      DT36                ;$ERRPC,PCJSR,TSTNM,MR,EC1,EC2
001554  067725                      DF36                ;0,0,0,0,0,0

                            ;*ITEM34
001556  061202                      EM34                ;ZERO DETECT BIT NOT HIGH WHEN THE
                                                        ;32 BIT ECC REGISTER HAS ITS 21 BITS
                                                        ;OF ZEROS
                                                        ;ERROR PRINTOUT WILL CONTINUE TILL
                                                        ;ZERO DETECT BIT IS HIGH
001560  066356                      DH36                ;PC
                                                        ;PC OF JSR
                                                        ;TEST NUMBER
                                                        ;RHMR
                                                        ;POSITION REG.
                                                        ;PATTERN REGISTER
001562  067454                      DT36                ;$ERRPC,PCJSR,TSTNM,MR,EC1,EC2
001564  067725                      DF36                ;0,0,0,0,0,0


                            ;*ITEM35
001566  061275                      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
001570  066515                      DH37                ;PC
```

```
                                                        ;TEST NUMBER
                                                        ;ECC POSITION
                                                        ;GOOD POSITION
                                                        ;GOOD ECC1
                                                        ;GOOD ECC2
                                                        ;ECC PATTERN
                                                        ;DATA ENVELOPE
                                                        ;M-CODE ZEROS

001572  067472                  DT37                  ;$ERRPC,TSTNM,EC1,POSITI,GECC1,GECC2,EC2,DATENV,ZCODE

001574  067733                  DF37                  ;0,0,0,0,0,0,0,0,0


                        ;*ITEM36
001576  061574                  EM36                  ;ON A READ COMMAND WITH NON CORRECTABLE
                                                      ;ERROR INSERTED DCK AND ECH SHOULD BE SET
001600  066153                  DH35                  ;PC
                                                      ;TEST NUMBER
                                                      ;GOOD ECC1
                                                      ;GOOD ECC2
                                                      ;PATTERN REGISTER
                                                      ;POSITION REGISTER
                                                      ;RHER1

001602  067432                  DT35                  ;$ERRPC,TSTNM,GECC1,GECC2,EC2,EC1,ER1

001604  067715                  DF35                  ;0,0,0,0,0,0,0

                        ;*ITEM37
001606  061705                  EM37                  ;PGE ERROR
001610  062676                  DH11                  ;PC
                                                      ;TEST NO
                                                      ;FAILING REG. ADDR
                                                      ;CONTENTS OF RHCS1
                                                      ;CONTENTS OF RHCS2
                                                      ;CONTENTS OF RHDS1
                                                      ;CONTENTS OF RHER1
001612  067052                  DT11                  ;$ERRPC,$TSTNM,$BDADR,CS1,CS2,DS1,ER1
001614  067544                  DF11                  ;0,0,0,0,0,0

                        ;*ITEM40
001616  062042                  EM40                  ;RHWC DID NOT = 0 AFTER A READ OR
                                                      ;WRITE HEADER AND DATA
001620  066733                  DH40                  ;PC
                                                      ;TEST NO
                                                      ;CONTENTS OF RHWC
001622  067516                  DT40                  ;$ERRPC,TSTNM,$BDDAT
001624  067744                  DF40                  ;0,0,0
```

m2

```
                        ;;*****************************************************************
                        ;*RH11/RH70 REGISTERS
                        ;;*****************************************************************


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


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


                        ;*CONTROL AND STATUS REGISTER 2 (RHCS2)

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

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


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


                        ;*CONTROL AND STATUS 1 REGISTER. (#00)

000001                  GO=     1               ;GO (BIT #0)
000100                  IE=     100             ;INTERRUPT ENABLE (BIT #6)
000200                  RDY=    200             ;READY (BIT #7)
000400                  A16=    400             ;HIGH ORDER UNIBUS BITS (BIT #8)
001000                  A17=    1000            ;HIGH ORDER UNIBUS BITS (BIT #9)
002000                  PSEL=   2000            ;PORT SELECT (BIT #10)
004000                  DVA=    4000            ;DEVICE AVAILABLE (BIT #11)
```

```
790          020000                  MCPE=   20000                   ;MASSBUSS PARITY ERROR (BIT #13)
791          040000                  TRE=    40000                   ;TRANSFER ERROR (BIT #14)
792          100000                  SC=     100000                  ;SPECIAL CONDITION (BIT #15)
793
794                                  ;*STATUS REGISTER (RHDS1) (#01)
795
796          000001                  DF5=    1                       ;DRIVE FORWARD 5"/SEC.  (BIT #0)
797          000002                  DFF20=  2                       ;DRIVE FORWARD 20"/SEC.  (BIT #1)
798          000004                  DIGB=   4                       ;DRIVE TO INNER GAVRD BAND (BIT #2)
799          000010                  GRV=    10                      ;GO REVERSE (BIT #3)
800          000020                  DL64=   20                      ;DIFFERENCE LESS THAN 64 (BIT #4)
801          000040                  DE1=    40                      ;DIFFERENCE EQUALS 1 (BIT #5)
802          000100                  VV=     100                     ;VOLUME VALID (BIT #6)
803          000200                  DRY=    200                     ;DRIVE READY (BIT #7)
804          000400                  DPR=    400                     ;DRIVE PRESENT (BIT #8)
805          001000                  PROG=   1000                    ;PROGRAMABLE (BIT #9)
806          002000                  LST=    2000                    ;LAST SECTOR TRANSFERRED (BIT #10)
807          004000                  WRL=    4000                    ;WRITE LOCK (BIT #11)
808          010000                  MOL=    10000                   ;MEDIUM ON-LINE (BIT #12)
809          020000                  PIP=    20000                   ;POSITIONING OPERATION IN PROGRESS (BIT #13)
810          040000                  ERR=    40000                   ;COMPOSIT ERROR.  (BIT #14)
811          100000                  ATA=    100000                  ;ATTENTION ACTIVE (BIT #15)
812
813                                  ;*ERROR REGISTER #01 (RHER1) (#02)
814          000001                  ILF=    1                       ;ILLEGAL FUNCTION (BIT #0)
815          000002                  ILR=    2                       ;ILLEGAL REGISTER (BIT #1)
816          000004                  RMR=    4                       ;REGISTER MODIFICATION REFUSED (BIT #2)
817          000010                  PAR=    10                      ;PARITY ERROR (BIT #3)
818          000020                  FER=    20                      ;FORMAT ERROR (BIT #4)
819          000040                  WCF=    40                      ;WRITE CLOCK FAIL (BIT #5)
820          000100                  ECH=    100                     ;ECC HARD ERROR (BIT #6)
821          000200                  HCE=    200                     ;HEADER COMPARE ERROR (BIT #7)
822          000400                  HCRC=   400                     ;HEADER CRC ERROR (BIT #8)
823          001000                  AOE=    1000                    ;ADDRESS OVERFLOW ERROR (BIT #9)
824          002000                  IAE=    2000                    ;INVALID ADDRESS ERROR (BIT #10)
825          004000                  WLE=    4000                    ;WRITE LOCK ERROR (BIT #11)
826          010000                  DTE=    10000                   ;DRIVE TIMING ERROR (BIT #12)
827          020000                  OPI=    20000                   ;OPERATION INCOMPLETE (BIT #13)
828          040000                  UNS=    40000                   ;DRIVE UNSAFE (BIT #14)
829          100000                  DCK=    100000                  ;DATA CHECK ERROR (BIT 15)
830
831                                  ;*MAINTAINABILITY REGISTER (RHMR)(#03)
832
833          000001                  DMD=    1                       ;DIAGINOSTIC MODE (BIT #0)
834          000002                  MCLK=   2                       ;MAINTAINABILITY CLOCK (BIT #1)
835          000004                  MINX=   4                       ;MAINTAINABILITY INDEX (BIT #2)
836          000010                  MSTCK=  10                      ;MAINTAINABILITY SECTOR CLOCK (BIT #3)
837          000020                  MRD=    20                      ;MAINTAINABILITY READ (BIT #4)
838          000040                  MWR=    40                      ;MAINTAINABILITY WRITE (BIT #5)
839          000200                  DENVL=  200                     ;DATA ENVELOPE (BIT #7)
840          000400                  ZER=    400                     ;ZERO DETECT (BIT #8)
841          001000                  DTSY=   1000                    ;MAINTAINABILITY SYNC DETECTED (BIT #9)
842
843                                  ;*ATTENTION SUMMARY PSEUDO-REGISTER (RHAS) (#04)
844
845          000001                  ATO=    1                       ;DEVICE 0 (BIT #0)
```

```
                        ERROR POINTER TABLE

000002          AT1=    2               ;DEVICE 1 (BIT #1)
000004          AT2=    4               ;DEVICE 2 (BIT #2)
000010          AT3=    10              ;DEVICE 3 (BIT #3)
000020          AT4=    20              ;DEVICE 4 (BIT #4)
000040          AT5=    40              ;DEVICE 5 (BIT #5)
000100          AT6=    100             ;DEVICE 6 (BIT #6)
000200          AT7=    200             ;DEVICE 7 (BIT #7)



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



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



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

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

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

000001          WCU=    1               ;WRITE CURRENT UNSAFE (BIT #0)
000002          CSF=    2               ;CURRENT SINK FAILURE (BIT #1)
000004          WSU=    4               ;WRITE SELECT UNSAFE (BIT #2)
000010          CSU=    10              ;CURRENT SWITCH UNSAFE (BIT #3)
000020          MSE=    20              ;MOTOR SEQUENCE ERROR (BIT #4)
000040          TDF=    40              ;TRANSITIONS DETECTON FAILURE (BIT #5)
000100          TUF=    100             ;TRANSITIONS UNSAFE (BIT #6)
000200          FEN=    200             ;FAILSAFE ENABLED (BIT #7)
000400          WRU=    400             ;WRITE READY UNSAFE (BIT #8)
```

```
                001000          MHS=    1000           ;MULTIPLE HEAD SELECT (BIT #9)
                002000          NHS=    2000           ;NO HEAD SELECTION (BIT #10)
                004000          IXE=    4000           ;INDEX ERROR (BIT #11)
                010000          VU30=   10000          ;30VOLT UNSAFE (BIT #12)
                020000          PLU=    20000          ;PLO UNSAFE (BIT #13)
                100000          ACU=    100000         ;ACUNSAFE (BIT #15)

                                ;*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)
000020                  PRE=    20              ;DISK PACK ROTATION ERROR (BIT #4)
000040                  ACL=    40              ;AC LOW (BIT #5)
000100                  DCL=    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  REGISTER ADDRESSES



                                ;*RP04 VECTOR ADDRESS

001626  000254                  RPVEC:  254                     ;RP04 VECTOR ADDRESS



                                ;*RP04/5/6 DISK I/O REGISTERS LOCATED IN THE RH11 CONTROLLER
                                ;*NOTE: THE CONTENTS OF THESE LOCATIONS WILL BE DIFFRENT
                                ;*      IF THE "CHANGE BASE ADDRESS" ROUTINE IS USED.
                                ;*      THIS ROUTINE STARTS AT LOCATION TAGED "BASECH"

001630  176722                  RHDB:   176722                  ;DATA BUFFER SEE NOTE ABOVE
001632  176702                  RHWC:   176702                  ;WORD COUNT SEE NOTE ABOVE
001634  176704                  RHBA:   176704                  ;BUS ADDRESS SEE NOTE ABOVE
001636  176710                  RHCS2:  176710                  ;CONTROL AND STATUS 2 SEE NOTE ABOVE



                                ;*RP04/5/6 DISK I/O REGISTERS LOCATED IN THE DEVICE CONTROL LOGIC (DCL)
                                ;*NOTE: THE CONTENTS OF THESE LOCATIONS WILL BE DIFFERENT
                                ;*      IF THE "CHANGE BASE ADDRESS ROUTINE IS USED.
                                ;*      THIS ROUTINE STARTS AT LOCATION TAGED "BASECH"

001640  176700                  RHCS1:  176700                  ;CONTROL AND STATUS 1 SEE NOTE ABOVE
001642  176714                  RHER1:  176714                  ;ERROR #1 SEE NOTE ABOVE
001644  176706                  RHDST:  176706                  ;DESIRED SECTOR/TRACK ADDRESS SEE NOTE ABOVE
001646  176740                  RHER2:  176740                  ;ERROR #2 SEE NOTE ABOVE
001650  176732                  RHOF:   176732                  ;OFFSET SEE NOTE ABOVE
001652  176734                  RHCA:   176734                  ;DESIRED CYLINDER ADDRESS SEE NOTE ABOVE
001654  176742                  RHER3:  176742                  ;ERROR #3 SEE NOTE ABOVE
001656  176716                  RHAS:   176716                  ;ATTENTION SUMMARY SEE NOTE ABOVE
001660  176724                  RHMR:   176724                  ;MAINTAINABILITY SEE NOTE ABOVE
001662  176712                  RHDS1:  176712                  ;DRIVE STATUS SEE NOTE ABOVE
001664  176726                  RHDT:   176726                  ;DRIVE TYPE SEE NOTE ABOVE
001666  176730                  RHSN:   176730                  ;SERIAL NUMBER SEE NOTE ABOVE
001670  176744                  RHEC1:  176744                  ;ECC POSITION SEE NOTE ABOVE
001672  176746                  RHEC2:  176746                  ;ECC PATTERN SEE NOTE ABOVE
001674  176720                  RHLA:   176720                  ;LOOK-AHEAD SEE NOTE ABOVE
001676  176736                  RHCC:   176736                  ;CURRENT CYLINDER ADDRESS SEE NOTE ABOVE

                                ;*ADDITIONAL REGISTERS LOCATED IN THE RH70 CONTROLLER LOGIC

001700  176750                  RHBAE:  176750                  ;BUS ADDRESS EXTENSION REGISTER
001702  176752                  RHCS3:  176752                  ;CONTROL AND STATUS REGISTER #3
```

```
1042
1043
1044                            ;*THE FOLLOWING LOCATIONS ARE RESERVED FOR REGISTER SAVES
1045                            ;*ANY TIME THERE IS AN ERROR ALL THESE WILL BE FILLED
1046                            ;*ONLY SOME MAY BE PRINTED BUT ALL WILL BE FILLED TRUE
1047                            ;*FOR THE TIME JUST AFTER THE "ERROR" ERROR COMMAND
1048
1049  001704  000000            DB:     0                       ;DATA BUFFER
1050  001706  000000            WC:     0                       ;WORD COUNT
1051  001710  000000            BA:     0                       ;BUS ADDRESS
1052  001712  000000            CS2:    0                       ;CONTROL AND STATUS 2
1053
1054
1055  001714  000000            CS1:    0                       ;CONTROL AND STATUS 1
1056  001716  000000            ER1:    0                       ;ERROR #1
1057  001720  000000            DST:    0                       ;DESIRED SECTOR/TRACK ADDRESS
1058  001722  000000            ER2:    0                       ;ERROR #2
1059  001724  000000            OF:     0                       ;OFFSET
1060  001726  000000            CA:     0                       ;DESIRED CYLINDER ADDRESS
1061  001730  000000            ER3:    0                       ;ERROR #3
1062  001732  000000            AS:     0                       ;ATTENTION SUMMARY
1063  001734  000000            MR:     0                       ;MAINTAINABILITY
1064  001736  000000            DS1:    0                       ;DRIVE STATUS
1065  001740  000000            DT:     0                       ;DRIVE TYPE
1066  001742  000000            SN:     0                       ;SERIAL NUMBER
1067  001744  000000            EC1:    0                       ;ECC POSITION
1068  001746  000000            EC2:    0                       ;ECC PATTERN
1069  001750  000000            LA:     0                       ;LOOK-AHEAD
1070  001752  000000            CC:     0                       ;CURRENT CYLINDER ADDRESS
```

```
1071
1072                                    ;*FLAGS & INTERNAL PROGRAM CONTROL WORDS
1073
1074  001754  000010                    UNITS:  .BLKW   8.              ;TABLE OF DRIVES PRESENT TO TEST
1075  001774  000000                    UNIT:   .WORD   0               ;UNIT UNDER TEST
1076  001776  000000                    NOUNIT: .WORD   0               ;NUMBER OF UNITS PRESENT
1077                                                                    ;USED TO KEEP TRACK OF UNIT UNDER TEST
1078  002000  000000                    NUNIT:  .WORD   0               ;USED TO DETERMINE IF THERE IS MORE
1079                                                                    ;THAN ONE UNIT
1080  002002  000000                    SELECT: .WORD   0               ;ALL ONES INDICATE UNIT TO BE SELECTED
1081  002004  000000                    UNITSL: .WORD   0               ;UNIT NO. SELECTED
1082
1083  002006  000000                    ERFLG$: 0                       ;ERROR FLAG
1084
1085  002010  000000                    SAVDT:  0                       ;SAVE DRIVE TYPE REGISTER
1086                                                                    ;FOR COMPARISON IN DRIVE CLEAR TEST
1087                                                                    ;AND RH INIT TEST
1088  002012  000000                    SAVSN:  0                       ;SAVE SERIAL NUMBER REGISTER
1089                                                                    ;FOR COMPARISON IN DRIVE CLEAR TEST
1090                                                                    ;AND RH INIT TEST
1091
1092  002014  000000                    PCJSR:  0                       ;SAVE PC OF JSR WHICH GAVE THE ERROR
1093
1094  002016  000000                    ATTENT: 0                       ;ATTENTION BIT FOR PRESENT UNIT
1095  002020  000000                    TOTALAT: 0                      ;TATAL ATTENTION BITS
1096
1097  002022  000000                    TMPILL: 0                       ;TEMPORARY ILLEGAL FUNCTION
1098
1099  002024  000000                    TSECC:  0                       ;FLAG TO SAY IF ECC TEST OR NOT
1100                                                                    ;WHEN =177777 IT IS AN ECC TEST
1101                                                                    ;WHEN =0IT IS NOT AN ECC TEST
1102
1103  002026  000000                    TESDTE: 0                       ;FLAG TO SAY IF DRIVE TIMING ERROR OR NOT
1104                                                                    ;WHEN = 177777 IT IS A DTE TEST
1105                                                                    ;WHEN = 0      IT IS NOT A DTE TEST
1106
1107  002030  000000                    TAGDTE: 0                       ;TEMPORARY TAG USED IN DRIVE TIMING
1108                                                                    ;ERROR TEST
1109
1110  002032  000000                    TSTNM:  0                       ;TEST NUMBER
1111
1112  002034  000000                    FIRST:  0                       ;IF ZERO WILL TYPE HEADER
1113
1114  002036  000000                    RP06:   0                       ;IF 0 PROGRAM WILL TREAT DRIVE AS RP04
1115
1116  002040  000000                    RH70:   0                       ;IF 1 PROGRAM IS RUNNING ON RH70
1117                                                                    ;IF 0 PROGRAM IS ON AN RH11
```

```
1118
1119
1120                                    ;*FUNCTION EQUATES
1121
1122                                    ;*TABLE OF FUNCTIONS FOR RHCS1, THEN "GO" BIT HAS TO BE SET
1123
1124  002042                    FUTABL:
1125  002042  000000           NOPERA: 0                       ;NO OPERATION
1126  002044  000002           UNLOAD: 2                       ;UNLOAD (STAND BY)
1127  002046  000006           RECALI: 6                       ;RECALIBRATE
1128  002050  000010           DCLEAR: 10                      ;DRIVE CLEAR
1129  002052  000012           RELEAS: 12                      ;RELEASE (DUAL-PORT OPERATION)
1130  002054  000030           SERCH:  30                      ;SEARCH COMMAND
1131  002056  000050           WRCHEK: 50                      ;WRITE CHECK DATA
1132  002060  000052           WRCHDT: 52                      ;WRITE CHECK HEADER AND DATA
1133  002062  000060           WRIDAT: 60                      ;WRITE DATA
1134  002064  000062           WRIFOR: 62                      ;WRITE HEADER AND DATA (FORMAT)
1135  002066  000070           READAT: 70                      ;READ DATA
1136  002070  000072           REFOR:  72                      ;READ HEADER AND DATA
1137  002072  000004           SEECOM: 4                       ;SEEK COMMAND
1138  002074  000014           OFSETC: 14                      ;OFFSET COMMAND
1139  002076  000016           RETCL:  16                      ;RETURNTO CENTERLINE
1140  002100  000022           PKACK:  22                      ;PACK ACKNOWLEDGE
1141  002102  000020           READIN: 20                      ;READ IN
1142  002104  000000           ILLEGL: .WORD                   ;COMPUTED ILLEGAL FUNCTION
1143
1144
1145                                    ;*DATA BUFFERS FOR READ WRITE
1146
1147
1148  002110  000422           WRFROM: .BLKW   274.            ;WRITE FROM THIS BUFFER
1149  003154  000422           REINTO: .BLKW   274.            ;READ INTO THIS BUFFER
1150
1151
1152
1153                                    ;*TABLE FOR ATTENTION BITS
1154                                    ;*ATTENTION TABLE
1155
1156  004220     001     002     004  ATABLE: .BYTE   1,2,4,10,20,40,100,200
1157  004223     010     020     040
1158  004226     100     200
1159
```

```
1160
1161                                            .SBTTL
1162                                            .SBTTL  ***DIAGNOSTIC CODE***
1163                                            .SBTTL
1164
1165                                            .SBTTL  SETUP TESTS
1166
1167  004230  012737  177777  002002  BEGIN2: MOV     #-1,@#SELECT    ;SELECT UNIT
1168  004236  000402                  BR      START
1169  004240  005037  002002          BEGIN:  CLR     @#SELECT        ;DO NOT SELECT UNIT
1170                                                                  ;NORMAL RUN
1171
1172  004244                          START:
1173  004244  000005                          RESET
1174                                  .SBTTL  INITIALIZE THE COMMON TAGS
1175                                  ;;CLEAR THE COMMON TAGS ($CMTAG) AREA
1176  004246  012706  001100                  MOV     #$CMTAG,R6      ;;FIRST LOCATION TO BE CLEARED
1177  004252  005026                          CLR     (R6)+           ;;CLEAR MEMORY LOCATION
1178  004254  022706  001140                  CMP     #SWR,R6 ;;DONE?
1179  004260  001374                          BNE     .-6             ;;LOOP BACK IF NO
1180  004262  012706  001000                  MOV     #STACK,SP       ;;SETUP THE STACK POINTER
1181                                  ;;INITIALIZE A FEW VECTORS
1182  004266  012737  053714  000020          MOV     #$SCOPE,@#IOTVEC ;;IOT VECTOR FOR SCOPE ROUTINE
1183  004274  012737  000340  000022          MOV     #340,@#IOTVEC+2 ;;LEVEL 7
1184  004302  012737  056132  000030          MOV     #$ERROR,@#EMTVEC ;;EMT VECTOR FOR ERROR ROUTINE
1185  004310  012737  000340  000032          MOV     #340,@#EMTVEC+2 ;;LEVEL 7
1186  004316  012737  056702  000034          MOV     #$TRAP,@#TRAPVEC ;;TRAP VECTOR FOR TRAP CALLS
1187  004324  012737  000340  000036          MOV     #340,@#TRAPVEC+2;LEVEL 7
1188  004332  012737  056772  000024          MOV     #$PWRDN,@#PWRVEC ;;POWER FAILURE VECTOR
1189  004340  012737  000340  000026          MOV     #340,@#PWRVEC+2 ;;LEVEL 7
1190  004346  005037  001212                  CLR     $TIMES          ;;INITIALIZE NUMBER OF ITERATIONS
1191  004352  005037  001214                  CLR     $ESCAPE         ;;CLEAR THE ESCAPE ON ERROR ADDRESS
1192  004356  112737  000001  001115          MOVB    #1,$ERMAX       ;;ALLOW ONE ERROR PER TEST
1193  004364  012737  004364  001106          MOV     #.,$LPADR       ;;INITIALIZE THE LOOP ADDRESS FOR SCOPE
1194  004372  012737  004372  001110          MOV     #.,$LPERR       ;;SETUP THE ERROR LOOP ADDRESS
1195                                  ;;SIZE FOR A HARDWARE SWITCH REGISTER. IF NOT FOUND OR IT IS
1196                                  ;;EQUAL TO A "-1", SETUP FOR A SOFTWARE SWITCH REGISTER.
1197  004400  013746  000004                  MOV     @#ERRVEC,-(SP)  ;;SAVE ERROR VECTOR
1198  004404  012737  004440  000004          MOV     #64$,@#ERRVEC   ;;SET UP ERROR VECTOR
1199  004412  012737  177570  001140          MOV     #DSWR,SWR       ;;SETUP FOR A HARDWARE SWICH REGISTER
1200  004420  012737  177570  001142          MOV     #DDISP,DISPLAY  ;;AND A HARDWARE DISPLAY REGISTER
1201  004426  022777  177777  174504          CMP     #-1,@SWR        ;;TRY TO REFERENCE HARDWARE SWR
1202  004434  001012                          BNE     66$             ;;BRANCH IF NO TIMEOUT TRAP OCCURRED
1203                                                                  ;;AND  THE HARDWARE SWR IS NOT = -1
1204  004436  000403                          BR      65$             ;;BRANCH IF NO TIMEOUT
1205  004440  012716  004446          64$:    MOV     #65$,(SP)       ;;SET UP FOR TRAP RETURN
1206  004444  000002                          RTI
1207  004446  012737  000176  001140  65$:    MOV     #SWREG,SWR      ;;POINT TO SOFTWARE SWR
1208  004454  012737  000174  001142          MOV     #DISPREG,DISPLAY
1209  004462  012637  000004          66$:    MOV     (SP)+,@#ERRVEC  ;;RESTORE ERROR VECTOR
1210
1211
1212
1213
1214  004466  012737  000000  177776  START1: MOV     #0,PS           ;SET PROCESSOR STATUS TO 0
1215  004474  012777  053632  175124          MOV     #RPVECT,@RPVEC  ;THIS IS FOR UNTIMELY DRIVE INTERRUPTS
```

```
1216  004502  004737  054672                  JSR     PC,@#$TKINT     ;INITIALIZE THE TTY KEYBOARD
1217  004506  005737  002034                  TST     @#FIRST         ;IS THIS FIRST TIME ROUND ?
1218  004512  001001                          BNE     1$              ;SKIP HEADER IF NOT
1219  004514  000402                          BR      2$              ;DO HEADER IF SO
1220
1221  004516  000137  005326          1$:     JMP     @#SND1          ;SKIP OVERALL PROGRAM HEADER
1222  004522                          2$:
1223  004522  104401  004530                  TYPE    ,65$            ;;TYPE ASCIZ STRING
1224  004526  000434                          BR      64$             ;;GET OVER THE ASCIZ
1225                                  ;;65$:  .ASCIZ  <15><12>?RP04/5/6 DISKLESS CONTROLLER TEST - PART I - DZRJG-A?
1226  004620                          64$:
1227
1228  004620  104401  004626                  TYPE    ,67$            ;;TYPE ASCIZ STRING
1229  004624  000417                          BR      66$             ;;GET OVER THE ASCIZ
1230                                  ;;67$:  .ASCIZ  <15><12>/REVISION DATE: 21-MAR-76/<15><12>
1231  004664                          66$:
1232
1233  004664  104401  004672                  TYPE    ,69$            ;;TYPE ASCIZ STRING
1234  004670  000433                          BR      68$             ;;GET OVER THE ASCIZ
1235                                  ;;69$:  .ASCIZ  <15><12>/ALL DCL'S UNDER TEST MUST BE LOCKED ON CORRECT PORT/
1236  004760                          68$:
1237  004760  104401  004766                  TYPE    ,71$            ;;TYPE ASCIZ STRING
1238  004764  000427                          BR      70$             ;;GET OVER THE ASCIZ
1239                                  ;;71$:  .ASCIZ  <15><12>/IF CHANGES ARE REQUIRED ON PORT SWITCH THEN/
1240  005044                          70$:
1241  005044  104401  005052                  TYPE    ,73$            ;;TYPE ASCIZ STRING
1242  005050  000430                          BR      72$             ;;GET OVER THE ASCIZ
1243                                  ;;73$:  .ASCIZ  <15><12>/A CYCLE UP SEQUENCE IS REQUIRED FOR STROBING/
1244  005132                          72$:
1245  005132  104401  005140                  TYPE    ,75$            ;;TYPE ASCIZ STRING
1246  005136  000415                          BR      74$             ;;GET OVER THE ASCIZ
1247                                  ;;75$:  .ASCIZ  <15><12>/THE PORT SELECT FLOP/<15><12>
1248  005172                          74$:
1249  005172  104401  005200                  TYPE    ,77$            ;;TYPE ASCIZ STRING
1250  005176  000430                          BR      76$             ;;GET OVER THE ASCIZ
1251                                  ;;77$:  .ASCIZ  <15><12>/ALL DCL'S NOT UNDER TEST MUST BE SWITCHED OFF/
1252  005260                          76$:
1253  005260  104401  005266                  TYPE    ,79$            ;;TYPE ASCIZ STRING
1254  005264  000420                          BR      78$             ;;GET OVER THE ASCIZ
1255                                  ;;79$:  .ASCIZ  <15><12>/OR LOCKED ON THE OTHER PORT/<15><12>
1256  005326                          78$:
1257
1258  005326  012737  177777  002034  SND1:   MOV     #-1,@#FIRST     ;NEXT TIME DO NOT GIVE HEADER
1259
1260                                  .SBTTL  GET VALUE FOR SOFTWARE SWITCH REGISTER
1261  005334  005737  000042                  TST     @#42            ;;ARE WE RUNNING UNDER XXDP/ACT?
1262  005340  001006                          BNE     64$             ;;BRANCH IF YES
1263  005342  023727  001140  000176          CMP     SWR,#SWREG      ;;SOFTWARE SWITCH REG SELECTED?
1264  005350  001005                          BNE     65$             ;;BRANCH IF NO
1265  005352  104406                          GTSWR                   ;;GET SOFT-SWR SETTINGS
1266  005354  000403                          BR      65$
1267  005356  112737  000001  001134  64$:    MOVB    #1,$AUTOB       ;;SET AUTO-MODE INDICATOR
1268  005364                          65$:
1269
1270  005364  032777  010000  173546  RH70CK: BIT     #SW12,@SWR      ;LOOK TO SEE IF USING RH70
1271  005372  001403                          BEQ     3$              ;IF SW12 = 0, SKIP NEXT
```

K3

```
1272  005374  012737  177777  002040          MOV     #-1,@#RH70      ;IF SW12 = 1, CU IS AN RH70
1273
1274  005402  005737  002002          3$:     TST     @#SELECT        ;WAS IT A 200 START
1275  005406  001434                          BEQ     TST1            ;SKIP NEXT IF STARTING FROM 200 ------------)
1276  005410  104401  005416                  TYPE    ,65$            ;;TYPE ASCIZ STRING
1277  005414  000422                          BR      64$             ;;GET OVER THE ASCIZ
1278                                  ;;65$:  .ASCIZ  <15><12>/SELECT UNIT NUMBER TO BE TESTED ?/
1279  005462                          64$:
1280  005462  104412                          RDOCT
1281  005464  042716  177770                  BIC     #177770,(SP)    ;ONLY KEEP LAST 3 BITS
1282  005470  011637  001774                  MOV     (SP),@#UNIT     ;SAVE UNIT TO BE TESTED
1283  005474  012637  002004                  MOV     (SP)+,@#UNITSL  ;SAVE UNIT TO BE TESTED
1284
1285
```

```
1286
1287
1288                                    ;;*****************************************************************
1289                                    ;*TEST 1         REFERENCE EACH REGISTER
1290
1291                                    ;**             REFERENCE EACH REGISTER BY A MOVE INSTRUCTION
1292
1293                                    ;;*****************************************************************
1294  005500  000004                    TST1:   SCOPE
1295  005502  012737  000001  001212            MOV     #1,$TIMES       ;;DO 1 ITERATION
1296  005510  012706  001000                    MOV     #STACK, SP      ;SET UP STACK POINTER
1297  005514  012737  000001  002032            MOV     #TTNO,@#TSTNM   ;THIS SAVES TEST NUMBER
1298
1299  005522  012737  056142  000030            MOV     #REGSA1,@#EMTVEC;ERROR VECTOR SO THAT
1300                                                                    ;NO REGISTERS ARE SAVED
1301  005530  012737  005556  000004            MOV     #2$,@#ERRVEC    ;SET UP FOR BUS TIMEOUT
1302  005536  012700  000024                    MOV     #24,R0          ;THERE ARE 24 REG TO TEST
1303  005542  012701  001630                    MOV     #RHDB,R1        ;R1 NOW HAS ADDR OF ADDR OF FIRST REG.
1304  005546  013102                    1$:     MOV     @(R1)+,R2       ;READ HARDWARE REG.
1305  005550  005300                            DEC     R0              ;COUNT DOWN
1306  005552  001375                            BNE     1$              ;BRANCH IF 24 NOT DONE
1307  005554  000471                            BR      3$              ;BRANCH IF 24 DONE
1308  005556  012737  000006  000004    2$:     MOV     #ERRVEC+2,@#ERRVEC ;RESTORE TRAP CATCHER
1309  005564  022626                            CMP     (SP)+,(SP)+     ;CLEAN STACK
1310  005566  016137  177776  001200            MOV     -2(R1), $TMP1   ;STORE FAILING REG ADDR
1311  005574  104015                            ERROR   15              ;REGISTER NON EXISTANT
1312  005576  032777  020000  173334            BIT     #SW13,@SWR      ;INHIBIT ERROR PRINTOUT ?
1313  005604  001053                            BNE     4$              ;BRANCH IF YES
1314
1315  005606  104401  005614                    TYPE    ,65$            ;;TYPE ASCIZ STRING
1316  005612  000431                            BR      64$             ;;GET OVER THE ASCIZ
1317                                    ;;65$:  .ASCIZ  <15><12>/IF BASE ADDRESS IS TO BE CHANGED HALT PROGRAM /
1318  005676                            64$:
1319  005676  104401  005704                    TYPE    ,67$            ;;TYPE ASCIZ STRING
1320  005702  000411                            BR      66$             ;;GET OVER THE ASCIZ
1321                                    ;;67$:  .ASCIZ  <15><12>/AND RESTART AT /
1322  005726                            66$:
1323
1324  005726  012746  046160                    MOV     #BASECH,-(SP)   ;GET READY TO TYPE STARTING ADDRESS
1325                                                                    ;OF "CHANGE OF BASE ADDRESS" ROUTINE
1326  005732  104402                            TYPOC
1327
1328  005734  000137  041234            4$:     JMP     @#$EOP          ;GO TO END OF PROGRAM ---------------->
1329
1330  005740  012737  056132  000030    3$:     MOV     #$ERROR,@#EMTVEC;RESTORE ERROR VECTOR
1331                                                                    ;SO THAT REGISTERS ARE SAVED
1332  005746  012737  000006  000004            MOV     #ERRVEC+2,@#ERRVEC ;RESTORE TRAP CATCHER
1333
```

```
1334
1335
1336
1337                                    ;;*****************************************************************
1338                                    ;*TEST 2        PHCS2-CONTROL AND STATUS 2
1339
1340                                    ;**     THIS PARTIALLY TESTS RHCS2 TO ENABLE DETERMINATION
1341                                    ;**     OF THE NUMBER OF DRIVES PRESENT
1342
1343                                    ;;*****************************************************************
1344  005754  000004                   TST2:   SCOPE
1345  005756  012737  000001  001212           MOV     #1,$TIMES       ;;DO 1 ITERATION
1346  005764  012706  001000           MOV     #STACK,SP       ;RESET STACK
1347  005770  012737  000002  002032           MOV     #TTNO,@#TSTNM   ;THIS SAVES TEST NUMBER
1348
1349                                    ;*CHECK TO SEE IF THE PROGRAM IS RUNNING WITH AN RH70
1350
1351  005776  005737  002040           TST     @#RH70          ;TEST FLAG FOR RH70 CONTROLLER
1352  006002  001402                   BEQ     30$             ;IF FLAG = 1, THIS TEST IS SKIPPED
1353  006004  000137  006040           JMP     TST3            ;JUMP TO NEXT TEST ------------------)
1354  006010                           30$:                    ;IF FLAG = 0, DO THIS TEST
1355
1356
1357  006010  013737  010656  006030           MOV     @#PRCS2+12,@#UN
1358  006016  013737  001636  006032           MOV     @#RHCS2,@#UN+2
1359  006024  004537  042706           JSR     R5,@#BITST      ;TEST BITS IN REGISTER
1360  006030  000000                   UN:     .WORD   0               ;ONLY THESE BITS TESTED FOR READ/WRITE
1361  006032  000000                           .WORD   0               ;ADDRESS OF REG. BEING TESTED
1362  006034  104001                           ERROR   1               ;IN CORRECT DATA RECEIVED
1363  006036  000207                           RTS     PC              ;RETURN TO BLT3 ROUTINE
1364
1365
1366
1367
1368
1369
1370                                    ;;*****************************************************************
1371                                    ;*TEST 3        PARTIAL TEST OF RHAS FOR UNIT NUMBERS PRESENT
1372                                    ;;*****************************************************************
1373  006040  000004                   TST3:   SCOPE
1374  006042  012737  000001  001212           MOV     #1,$TIMES       ;;DO 1 ITERATION
1375
1376  006050  012737  000003  002032           MOV     #TTNO,@#TSTNM   ;THIS SAVES TEST NUMBER
1377
1378  006056  013701  001656           MOV     @#RHAS,R1       ;R1 HAS ADDRESS OF RHAS
1379  006062  012711  177777           MOV     #-1,@R1         ;THIS CLEARS RHAS (SURPRISED!)
1380  006066  011137  001126           MOV     @R1,@#$BDDAT    ;TEST DATA
1381  006072  105737  001126           TSTB    @#$BDDAT
1382  006076  001405                   BEQ     TST4            ;BRANCH IF GOOD
1383  006100  005037  001124           CLR     @#$GDDAT        ;GOOD DATA
1384  006104  010137  042204           MOV     R1,@#REGADR     ;FAILING REG. RHAS
1385  006110  104001                   ERROR   1               ;RHAS DOES NOT CLEAR
1386                                                            ;BY WRITING ONES IN
```

```
1387
1388
1389                                    ;;*****************************************************************
1390                                    ;*TEST 4        TEST FOR DRIVES PRESENT USING RHAS AND RHCS2
1391                                    ;;*****************************************************************
1392  006112  000004                    TST4:   SCOPE
1393  006114  012737  000001  001212            MOV     #1,$TIMES       ;;DO 1 ITERATION
1394
1395  006122  000005                            RESET                   ;START WITH AN INIT
1396  006124  012737  000004  002032            MOV     #TTNO,@#TSTNM   ;THIS SAVES TEST NUMBER
1397  006132  004737  054672                    JSR     PC,@#$TKINT     ;INITILIZE TTY KEYBOARD
1398  006136  032777  020000  172774            BIT     #SW13,@SWR      ;INHIBIT ERROR TYPEOUT?
1399  006144  001147                            BNE     4$              ;BRANCH IF YES
1400
1401  006146  104401  006154                    TYPE    ,65$            ;;TYPE ASCIZ STRING
1402  006152  000430                            BR      64$             ;;GET OVER THE ASCIZ
1403                                    ;;65$:  .ASCIZ  <15><12><15><12>/LOOKING AT RHAS - DRIVES ASSUMED PRESENT/<15><12>
1404  006234                            64$:
1405  006234  104401  006242                    TYPE    ,67$            ;;TYPE ASCIZ STRING
1406  006240  000427                            BR      66$             ;;GET OVER THE ASCIZ
1407                                    ;;67$:  .ASCIZ  <15><12>/*********************************************/
1408  006320                            66$:
1409  006320  104401  006326                    TYPE    ,69$            ;;TYPE ASCIZ STRING
1410  006324  000425                            BR      68$             ;;GET OVER THE ASCIZ
1411                                    ;;69$:  .ASCIZ  <15><12>/   THIS MUST BE VERIFIED BY OPERATOR !!/
1412  006400                            68$:
1413  006400  104401  006406                    TYPE    ,71$            ;;TYPE ASCIZ STRING
1414  006404  000427                            BR      70$             ;;GET OVER THE ASCIZ
1415                                    ;;71$:  .ASCIZ  <15><12>/*********************************************/
1416  006464                            70$:
1417
1418  006464  013701  001656            4$:     MOV     @#RHAS,R1       ;LOAD R1 WITH ADDR. OF RHAS
1419  006470  013702  001636                    MOV     @#RHCS2,R2      ;LOAD R2 WITH ADDR. OF RHCS2
1420  006474  005012                            CLR     @R2             ;CLEAR RHCS2 (ADDRESS UNIT #0)
1421  006476  012700  000010                    MOV     #8.,R0          ;INITIALIZE DRIVE COUNTER
1422  006502  013704  001642                    MOV     @#RHER1,R4      ;LOAD R4 WITH ADDR. OF RHER1
1423  006506  012714  177777            1$:     MOV     #-1,@R4         ;MOVE ERRORS INTO RHER1 OF UNIT ADDRESSED
1424  006512  005212                            INC     @R2             ;INCREMENT UNIT NO. (RHCS2)
1425  006514  005300                            DEC     R0              ;COUNT DOWN DRIVE COUNTER
1426  006516  001373                            BNE     1$              ;TEST AND DO NEXT UNIT IF 8 NOT DONE
1427
1428  006520  111137  002020                    MOVB    @R1,@#TOTALAT   ;SAVE ALL RESULTING ATTENTION BITS
1429                                            ;(USED IN DRIVE CLEAR TEST)
1430  006524  105037  002021                    CLRB    @#TOTALAT+1     ;CLEAR UPPER BYTE
1431  006530  105711                            TSTB    @R1             ;TEST RHAS FOR ANY DRIVES PRESENT
1432  006532  001402                            BEQ     2$              ;NONE RESPONDING - TYPE THE MESSAGE
1433  006534  000137  007106                    JMP     XE2             ;SOME THERE - GO FILL "UNITS" TABLE
1434
1435  006540  032777  020000  172372    2$:     BIT     #SW13,@SWR      ;INHIBIT ERROR TYPE OUT?
1436  006546  001402                            BEQ     3$              ;TYPE "NO DRIVES" MESSAGE IF NO
1437  006550  000137  007444                    JMP     SELTST          ;CHECK FOR SELECTED UNIT START AND LOAD
1438                                            ;"UNITS" TABLE WITH DESIRED DRIVE IF SO
1439
1440  006554                            3$:
1441  006554  104401  006562                    TYPE    ,73$            ;;TYPE ASCIZ STRING
1442  006560  000421                            BR      72$             ;;GET OVER THE ASCIZ
```

```
1443                                        ;;73$:  .ASCIZ  <15><12>/NO DRIVES PRESENT - RHAS = 0/<15><12>
1444  006624                                72$:
1445  006624  104401  006632                        TYPE    ,75$            ;;TYPE ASCIZ STRING
1446  006630  000430                        BR      74$             ;;GET OVER THE ASCIZ
1447                                        ;;75$:  .ASCIZ  <15><12>/WRITING ONES INTO RHER1 FOR ALL UNIT NUMBERS/
1448  006712                                74$:
1449  006712  104401  006720                        TYPE    ,77$            ;;TYPE ASCIZ STRING
1450  006716  000430                        BR      76$             ;;GET OVER THE ASCIZ
1451                                        ;;77$:  .ASCIZ  <15><12>/DOES NOT SET ANY BIT IN RHAS SO ABORT PROGRAM/
1452  007000                                76$:
1453  007000  104401  007006                        TYPE    ,79$            ;;TYPE ASCIZ STRING
1454  007004  000436                        BR      78$             ;;GET OVER THE ASCIZ
1455                                        ;;79$:  .ASCIZ  <15><12>/TO LOOP ON THIS TEST WO PRINTOUT, SET SWITCHES 13, 8 & 2/
1456  007102                                78$:
1457
1458  007102  000137  041234                        JMP     @#$EOP          ;GO OUT------------------------------>
1459
1460
1461                                        ;*SET UP DRIVES PRESENT TABLE
1462  007106                                XE2:
1463
1464  007106  012700  000010                2$:     MOV     #8.,R0          ;LOAD "UNITS" TABLE COUNTER
1465  007112  012703  001754                        MOV     #UNITS,R3       ;LOAD "UNITS" TABLE POINTER
1466  007116  012723  177777                3$:     MOV     #-1,(R3)+       ;PRESET 1ST TABLE BLOCK TO ALL ONES
1467  007122  005300                        DEC     R0              ;COUNT DOWN
1468  007124  001374                        BNE     3$              ;PRESET NEXT BLOCK IF 8 NOT DONE
1469
1470  007126  012703  001754                10$:    MOV     #UNITS,R3       ;RELOAD THE TABLE POINTER
1471  007132  005005                        CLR     R5              ;INITIALIZE UNIT NO. TO 0
1472  007134  005037  001776                        CLR     @#NOUNIT        ;NO. OF UNITS PRESENT
1473  007140  012700  000010                        MOV     #8.,R0          ;RELOAD THE TABLE COUNTER
1474  007144  011137  001176                        MOV     @R1,@#$TMP0     ;ADDR OF RHAS INTO TEMPORARY STORAGE
1475  007150  006037  001176                4$:     ROR     @#$TMP0         ;SET CARRY IF 0 BIT = 1 (UNIT ATTEN.)
1476  007154  103120                        BCC     5$              ;CHECK NEXT UNIT IF ONE NOT IN BIT 0
1477
1478  007156  010577  172454                11$:    MOV     R5,@RHCS2       ;INSERT UNIT NO. INTO RHCS2 UNIT ADDR.
1479  007162  022777  024020  172474                CMP     #24020,@RHDT    ;READ RHDT - IS IT A DUAL PORT RP04 ?
1480  007170  001503                        BEQ     6$              ;YES...TYPE THE UNIT NO.
1481  007172  022777  020020  172464                CMP     #20020,@RHDT    ;READ RHDT - IS IT A SINGLE PORT RP04 ?
1482  007200  001477                        BEQ     6$              ;YES...TYPE THE UNIT NO.
1483
1484
1485
1486                                        ;;*****************************************************************
1487  007202  022777  024021  172454                CMP     #24021,@RHDT    ;DUAL PORT RP05 ?
1488  007210  001473                        BEQ     6$              ;TYPE UNIT NO. IF SO
1489  007212  022777  020021  172444                CMP     #20021,@RHDT    ;SINGLE PORT RP05 ?
1490  007220  001467                        BEQ     6$              ;TYPE NO. IF SO
1491
1492  007222  022777  024022  172434                CMP     #24022,@RHDT    ;READ RHDT - IS IT A DUAL PORT RP06 ?
1493  007230  001463                        BEQ     6$              ;YES...TYPE THE UNIT NO.
1494  007232  022777  020022  172424                CMP     #20022,@RHDT    ;READ RHDT - IS IT A SINGLE PORT RP06 ?
1495  007240  001457                        BEQ     6$              ;YES...TYPE THE UNIT NO.
1496                                        ;;*****************************************************************
1497
1498
```

C4

```
1499
1500                                        ;*NO...IT'S NOT AN RP04/RP05/RP06 DEVICE SO TYPE
1501                                        ;*OUT THE DEVICE TYPE
1502
1503  007242  104401  007250                TYPE    ,65$            ;;TYPE ASCIZ STRING
1504  007246  000410                       BR      64$             ;;GET OVER THE ASCIZ
1505                              ;;65$:    .ASCIZ  <15><12>/UNIT NUMBER /
1506  007270                    64$:
1507
1508  007270  010546                       MOV     R5,-(SP)        ;PUT THE UNIT NUMBER ON STACK
1509  007272  104405                       TYPDS                   ;TYPE IT
1510  007274  104401  007302               TYPE    ,67$            ;;TYPE ASCIZ STRING
1511  007300  000406                       BR      66$             ;;GET OVER THE ASCIZ
1512                              ;;67$:    .ASCIZ  /,  RHDT = /    .
1513  007316                    66$:
1514  007316  017746  172342               MOV     @RHDT,-(SP)     ;PUT RHDT ON THE STACK
1515  007322  104402                       TYPOC                   ;TYPE IT
1516  007324  104401  007332               TYPE    ,69$            ;;TYPE ASCIZ STRING
1517  007330  000422                       BR      68$             ;;GET OVER THE ASCIZ
1519                              ;;69$:    .ASCIZ  ? - NOT AN RP04/RP05/RP06 DEVICE !!?
1519  007376                    68$:
1520  007376  000407                       BR      5$              ;UNIT NOT AN RP04/RP05/RP06 SO TEST NEXT ONE
1521
1522  007400  010523              6$:      MOV     R5,(R3)+        ;LOAD TABLE POSITION AND INCR IT
1523  007402  104401  001223               TYPE    ,$CRLF          ;CRLF
1524  007406  010546                       MOV     R5,-(SP)        ;PUT UNIT NO. ON THE STACK
1525  007410  104405                       TYPDS                   ;TYPE THE UNIT NO.
1526  007412  005237  001776               INC     @#NOUNIT        ;INCR THE TOTAL NO. OF UNITS
1527
1528  007416  005205              5$:      INC     R5              ;"RHCS2" UNIT ADDRESS
1529  007420  005300                       DEC     R0              ;DRIVE COUNTER DOWN ONE
1530  007422  001252                       BNE     4$              ;TEST AND DO NEXT UNIT IF 8 NOT DONE
1531
1532  007424  013737  001754  001774  12$: MOV     @#UNITS,@#UNIT  ;SET UNIT NO. TO FIRST ONE FOUND/OR 0
1533  007432  013737  001776  002000       MOV     @#NOUNIT,@#NUNIT ;SAVE NO. OF UNITS
1534  007440  005337  002000               DEC     @#NUNIT         ;IF NUNIT = 0 THEN ONLY ONE UNIT
1535                                                                ;IF NUNIT > 0 THEN MORE THAN ONE UNIT
1536
1537  007444  005737  002002        SELTST: TST     @#SELECT        ;STARTING ADDRESS 200 ?
1538  007450  001403                       BEQ     TST5            ;BRANCH IF STARTING FROM 200
1539  007452  013737  002004  001774       MOV     @#UNITSL,@#UNIT ;CHANGE UNIT NUMBER TO SELECTED ONE
1540
```

```
1541
1542
1543                                    ;;*****************************************************************
1544                                    ;*TEST 5         TYPE SERIAL NUMBER AND DRIVE TYPE
1545
1546                                    ;**     SET APPROPRIATE ATTENTION BIT OF UNIT UNDER TEST IN
1547                                    ;**     'ATTENT' AND TYPE THE UNIT UNDER TEST
1548
1549                                    ;**     READ SERIAL NUMBER REGISTER AND DRIVE TYPE REGISTER,
1550                                    ;**     TYPE THEM OUT AND PROCEED
1551
1552                                    ;**     TO LOOP HERE SET SWITCH 8 AND THIS TEST NO. AND RESTART
1553
1554                                    ;;*****************************************************************
1555  007460  000004                    TST5:   SCOPE
1556  007462  012737  000001  001212            MOV     #1,$TIMES       ;;DO 1 ITERATION
1557  007470  012737  010206  001106            MOV     #1$,$LPADR      ;;SET SCOPE LOOP ADDRESS
1558
1559  007476  012737  000005  002032            MOV     #TTNO,@#TSTNM   ;THIS SAVES TEST NUMBER
1560  007504  004737  042534                    JSR     PC,@#CLDISK     ;FILL UNIT NO.
1561  007510  005037  002016                    CLR     @#ATTENT        ;CLEAR
1562
1563                                            ;*TEST FOR UNIT #0
1564
1565  007514  005737  001774                    TST     @#UNIT          ;IS UNIT #0 NEXT IN THE UNITS TABLE ?
1566  007520  001022                            BNE     10$             ;IF NOT, TEST THIS UNIT
1567  007522  012700  000041                    MOV     #41,R0          ;IF SO, CHECK THE LOAD MEDIA LOCATION
1568  007526  122710  000011                    CMPB    #11,(R0)        ;WAS IT AN RP04/5/6 ?
1569  007532  001015                            BNE     10$             ;NO...GO AHEAD WITH TESTING UNIT #0
1570  007534  005737  002002                    TST     @#SELECT        ;WAS UNIT #0 SELECTED ?
1571                                                                    ;(IE. WAS IT A 210 START ?)
1572  007540  001012                            BNE     10$             ;IF SO...TEST IT
1573
1574                                            ;*INCREMENT THE UNITS TABLE TO NEXT DRIVE (IF ANY)
1575                                            ;*& DECREMENT THE "NOUNITS" PRESENT (TO BE TESTED)
1576
1577  007542  012700  001754                    MOV     #UNITS,R0       ;LOAD THE UNITS TABLE POINTER
1578  007546  005720                            TST     (R0)+           ;SELECT THE NEXT UNIT IN THE TABLE
1579                                                                    ;(DOUBLE INCREMENT THE POINTER, R0)
1580  007550  022710  177777                    CMP     #-1,(R0)        ;IS THERE ANOTHER TABLE ENTRY PRESENT ?
1581  007554  001404                            BEQ     10$             ;IF NOT (LOC = -1)...MUST USE UNIT #0
1582  007556  011037  001774                    MOV     (R0),@#UNIT     ;SET UP TO BE THE UNIT UNDER TEST
1583  007562  005337  001776                    DEC     @#NOUNITS       ;DECREMENT BECAUSE UNIT # 0 WONT'T BE TESTED
1584  007566  013700  001774            10$:    MOV     @#UNIT,R0       ;R0 CONTAINS UNIT NO.
1585
1586
1587                                            ;*SET UP THE PROPER DEVICE TYPE FLAG
1588
1589                                    ;;*****************************************************************
1590  007572  010077  172040                    MOV     R0,@RHCS2       ;SET UP UNIT ADDRESS
1591  007576  005037  002036                    CLR     @#RP06          ;CLEAR RP06 DEVICE TYPE FLAG
1592  007602  022777  024022  172054            CMP     #24022,@RHDT    ;DUAL PORT RP06 ?
1593  007610  001405                            BEQ     2$              ;YES...SET THE FLAG
1594  007612  022777  020022  172044            CMP     #20022,@RHDT    ;SINGLE PORT RP06 ?
1595  007620  001401                            BEQ     2$              ;YES..SET FLAG
1596  007622  000403                            BR      3$              ;NO...DON'T SET RP06 FLAG
```

```
                        T5      TYPE SERIAL NUMBER AND DRIVE TYPE

1597  007624  012737  177777  002036  2$:     MOV     #-1,@#RP06      ;SET IT
1598
1599  007632                          3$:                             ;ASSUME THE NEXT UNIT IS AN RP04
1600                                  ;;*************************************************************
1601
1602  007632  116037  004220  002016          MOVB    ATABLE(R0),@#ATTENT ;SET APPROPRIATE ATTENTION BIT
1603
1604  007640  104401  007646                  TYPE    ,65$            ;;TYPE ASCIZ STRING
1605  007644  000414                          BR      64$             ;;GET OVER THE ASCIZ
1606                                  ;;65$:  .ASCIZ  <15><12>/TESTING DRIVE NUMBER/
1607  007676                          64$:
1608  007676  013746  001774                  MOV     @#UNIT,-(SP)    ;UNIT NO. TO STACK
1609  007702  104405                          TYPDS                   ;TYPE  DRIVE NO.
1610  007704  104401  007712                  TYPE    ,67$            ;;TYPE ASCIZ STRING
1611  007710  000410                          BR      66$             ;;GET OVER THE ASCIZ
1612                                  ;;67$:  .ASCIZ  <15><12>/SERIAL NO. = /
1613  007732                          66$:
1614  007732  017746  171730                  MOV     @RHSN,-(SP)     ;;SAVE @RHSN FOR TYPEOUT
1615  007736  104402                          TYPOC                   ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
1616  007740  104401  007746                  TYPE    ,69$            ;;TYPE ASCIZ STRING
1617  007744  000410                          BR      68$             ;;GET OVER THE ASCIZ
1618                                  ;;69$:  .ASCIZ  <15><12>/DRIVE TYPE = /
1619  007766                          68$:
1620  007766  017746  171672                  MOV     @RHDT,-(SP)     ;;SAVE @RHDT FOR TYPEOUT
1621  007772  104402                          TYPOC                   ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
1622
1623                                  ;;*************************************************************
1624  007774  022777  024020  171662          CMP     #24020,@RHDT    ;DUAL PORT RP04 ?
1625  010002  001425                          BEQ     4$              ;TYPE ASCII MSG OUT
1626  010004  022777  020020  171652          CMP     #20020,@RHDT    ;SINGLE PORT RP04 ?
1627  010012  001421                          BEQ     4$              ;TYPE THE MESSAGE
1628
1629  010014  022777  024021  171642          CMP     #24021,@RHDT    ;DUAL PORT RP05 ?
1630  010022  001453                          BEQ     6$              ;TYPE MSG
1631  010024  022777  020021  171632          CMP     #20021,@RHDT    ;SINGLE PORT RP05 ?
1632  010032  001447                          BEQ     6$              ;TYPE MSG
1633
1634  010034  022777  024022  171622          CMP     #24022,@RHDT    ;DUAL PORT RP06 ?
1635  010042  001424                          BEQ     5$              ;TYPE MSG
1636  010044  022777  020022  171612          CMP     #20022,@RHDT    ;SINGLE PORT RP06 ?
1637  010052  001420                          BEQ     5$              ;TYPE MSG
1638  010054  000454                          BR      1$              ;DRIVE IS NOT AN RP04/RP05/RP06  - SO
1639                                                                  ;DON'T TYPE THE ASCII MESSAGE
1640
1641                                                                  ;-SHOULD NEVER HAPPEN AT THIS POINT
1642                                                                  ;UNLESS DRIVE GOT SICK WHILE TESTING
1643                                                                  ;WAS IN PROGRESS
1644  010056                          4$:
1645  010056  104401  010064                  TYPE    ,71$            ;;TYPE ASCIZ STRING
1646  010062  000413                          BR      70$             ;;GET OVER THE ASCIZ
1647                                  ;;71$:  .ASCIZ  <15><12>/DRIVE IS AN RP04/<15><12>
1648  010112                          70$:
1649  010112  000435                          BR      1$              ;SKIP NEXT MESSAGE
1650  010114                          5$:
1651  010114  104401  010122                  TYPE    ,73$            ;;TYPE ASCIZ STRING
1652  010120  000413                          BR      72$             ;;GET OVER THE ASCIZ
```

```
1653                                    ;;73$:  .ASCIZ  <15><12>/DRIVE IS AN RP06/<15><12>
1654  010150                            72$:
1655  010150  000416                           BR      1$                ;SKIP NEXT
1656  010152                            6$:
1657  010152  104401  010160                   TYPE    ,75$              ;;TYPE ASCIZ STRING
1658  010156  000413                           BR      74$               ;;GET OVER THE ASCIZ
1659                                    ;;75$:  .ASCIZ  <15><12>/DRIVE IS AN RP05/<15><12>
1660  010206                            74$:
1661                                    ;;**********************************************************************
1662
1663  010206  005777  171454           1$:     TST     @RHSN             ;READ SERIAL NO. AND DRIVE TYPE
1664  010212  005777  171446                   TST     @RHDT             ;THESE TWO ARE TO HELP SCOPE LOOPS
1665  010216  017737  171444  002012           MOV     @RHSN,@#SAVSN     ;SAVE TO CHECK IF CLR RHCS2 BIT 5 CLEARS ANY BITS
1666  010224  017737  171434  002010           MOV     @RHDT,@#SAVDT     ;SAVE TO CHECK IF CLR RHCS2 BIT 5 CLEARS ANY BITS
1667
1668
```

```
1669
1670                                    ;;*****************************************************************
1671                                    ;*TEST 6        CHECK MOL TO BE LOW
1672
1673                                    ;**     MAKE SURE THAT DRIVE IS OFF LINE BEFORE STARTING PROGRAM
1674                                    ;**     IF DRIVE IS ON LINE THEN AFTER TYPE OUT THE PROGRAM WILL
1675                                    ;**     HANG FOR EVER WAITING FOR DRIVE TO GO OFF LINE
1676
1677                                    ;;*****************************************************************
1678  010232  000004                    TST6:   SCOPE
1679  010234  012737  000006  002032            MOV     #TTNO,@#TSTNM   ;THIS SAVES TEST NUMBER
1680  010242  004737  042534                    JSR     PC,@#CLDISK     ;GIVE INITILIZE
1681  010246  032713  010000                    BIT     #MOL,@R3        ;CHECK MOL IN RHDS1
1682  010252  001551                            BEQ     TST7            ;BRANCH IF MOL LOW
1683  010254  104401  010262                    TYPE    ,65$            ;;TYPE ASCIZ STRING
1684  010260  000421                            BR      64$             ;;GET OVER THE ASCIZ
1685                                    ;;65$:  .ASCIZ  <15><12>/DRIVE IS ON LINE - MOL IS HIGH/
1686  010324                            64$:
1687  010324  104401  010332                    TYPE    ,67$            ;;TYPE ASCIZ STRING
1688  010330  000424                            BR      66$             ;;GET OVER THE ASCIZ
1689                                    ;;67$:  .ASCIZ  <15><12>/HIT STOP ON DRIVE TO GET IT OFF LINE/
1690  010402                            66$:
1691  010402  104401  010410                    TYPE    ,69$            ;;TYPE ASCIZ STRING
1692  010406  000430                            BR      68$             ;;GET OVER THE ASCIZ
1693                                    ;;69$:  .ASCIZ  <15><12>/PROGRAM WILL HANG TESTING MOL TILL MOL IS LOW/
1694  010470                            68$:
1695  010470  032713  010000           1$:     BIT     #MOL,@R3        ;CHECK MOL IN RHDS1
1696  010474  001375                            BNE     1$              ;BRANCH IF MOL IS HIGH
1697  010476  104401  010504                    TYPE    ,71$            ;;TYPE ASCIZ STRING
1698  010502  000435                            BR      70$             ;;GET OVER THE ASCIZ
1699                                    ;;71$:  .ASCIZ  <15><12>/GOOD - MOL IS NOW LOW . PROGRAM WILL NOW BE EXECUTED/<15><12>
1700  010576                            70$:
```

```
1701
1702                                            .SBTTL  REGISTER TESTS
1703
1704                                    ;;*******************************************************************
1705                                    ;*TEST 7         RHCS2 - CONTROL AND STATUS 2
1706                                    ;**             TEST LOADING AND READING OF ALL POSSIBLE BITS IN THE HARDWARE
1707                                    ;**             REGISTERS.  USE A PETTERN OF WALKING 0'S (-2,-3,-5 ETC.) AND
1708                                    ;**             WALKING 1'S (1,2,4,10 ETC)
1709                                    ;;*******************************************************************
1710  010576  000004                    TST7:   SCOPE
1711  010600  012737  000007  002032            MOV     #TTNO,@#TSTNM   ;THIS SAVES TEST NUMBER
1712
1713                                            ;*CHECK TO SEE IF THE PROGRAM IS RUNNING WITH AN RH70
1714
1715  010606  005737  002040            TST     @#RH70          ;TEST FLAG FOR RH70 CONTROLLER
1716  010612  001402                    BEQ     30$             ;IF FLAG = 1, THIS TEST IS SKIPPED
1717  010614  000137  010666            JMP     TST10           ;JUMP TO NEXT TEST ------------------)
1718  010620                            30$:                    ;IF FLAG = 0, DO THIS TEST
1719
1720  010620  004737  042534            JSR     PC,@#CLDISK     ;GIVE INITIALIZE
1721
1722
1723  010624  012706  001000            MOV     #STACK,SP       ;RESET STACK
1724  010630  012737  000007  002032            MOV     #TTNO,@#TSTNM   ;THIS SAVES TEST NUMBER
1725
1726  010636  013737  001636  010660            MOV     @#RHCS2,@#PRCS2+14      ;GET REGISTER ADDRESS
1727  010644  013777  001774  170764  PRCS2:   MOV     @#UNIT,@RHCS2   ;MOVE UNIT NO. UNDER TEST
1728  010652  004537  042206                    JSR     R5,BITST        ;TEST BITS IN REGISTER
1729  010656  020017                            .WORD   20017           ;ONLY THESE BITS ARE TESTED FOR READ/WRITE
1730  010660  176710                            .WORD   176710          ;ADDRESS OF REG. BEING TESTED
1731  010662  104001                            ERROR   1               ;INCORRECT DATA RECEIVED
1732  010664  000207                            RTS     PC              ;RETURN TO BLT3 ROUTINE
1733
1734
1735
1736
1737
1738
1739
1740                                    ;;*******************************************************************
1741                                    ;*TEST 10        RHCS1 - CONTROL AND STATUS 1 REGISTER
1742                                    ;**             TEST LOADING AND READING OF ALL POSSIBLE BITS IN THE HARDWARE
1743                                    ;**             REGISTERS.  USE A PATTERN OF WALKING 0'S (-2,-3,-5 ETC) AND
1744                                    ;**             WALKING 1'S (1,2,4,10 ETC)
1745                                    ;;*******************************************************************
1746  010666  000004                    TST10:  SCOPE
1747
1748  010670  012706  001000            MOV     #STACK,SP       ;RESET STACK
1749  010674  012737  000010  002032            MOV     #TTNO,@#TSTNM   ;THIS SAVES TEST NUMBER
1750
1751  010702  013737  001640  010724            MOV     @#RHCS1,@#PRCS1+14      ;GET REGISTER ADDRESS
1752  010710  013777  001774  170720  PRCS1:   MOV     @#UNIT,@RHCS2   ;MOVE UNIT NO. UNDER TEST
1753  010716  004537  042206                    JSR     R5,BITST        ;TEST BITS IN REGISTER
1754  010722  001476                            .WORD   1476            ;ONLY THESE BITS ARE TESTED FOR READ/WRITE
1755  010724  176700                            .WORD   176700          ;ADDRESS OF REG. BEING TESTED
1756  010726  104001                            ERROR   1               ;INCORRECT DATA RECEIVED
```

```
 1757  010730  000207                          RTS     PC              ;RETURN TO SLT3 ROUTINE
 1758
 1759
```

```
1760
1761
1762                                    ;;*****************************************************************
1763                                    ;*TEST 11       RHCS1 - BIT # 13 - MCPE
1764                                    ;**     THIS FORCES A MASS BUS CONTROL PARITY ERROR
1765                                    ;**     BY SETTING 'PAT', LOOKING FOR 'PAR', WRITING RHCS1
1766                                    ;**     AND READING RHER1
1767                                    ;;*****************************************************************
1768  010732  000004                    TST11:  SCOPE
1769
1770  010734  012706  001000                    MOV     #STACK,SP       ;RESET STACK
1771  010740  012737  000011  002032            MOV     #TTNO,@#TSTNM   ;THIS SAVES TEST NUMBER
1772  010746  004737  042534                    JSR     PC,@#CLDISK     ;INIT AND SET UNIT NUMBER AND DEVICE -
1773                                                                    ;CPU REG. CORRESPONDENCE (R1-R4)
1774
1775                                    ;*SET FORCED PARITY ERROR 'PAT'
1776  010752  052777  000020  170656            BIS     #PAT,@RHCS2     ;SET 'PAT' TO INVERT PARITY
1777                                                                    ;GENERATED
1778  010760  005077  170656                    CLR     @RHER1          ;WRITE DCL REGISTER USING BAD CONTROL PARITY
1779
1780                                    ;*WITH THIS PARITY ERROR NOTHING WILL BE READ TILL IT IS
1781                                    ;*CLEARED
1782
1783  010764  011137  001126                    MOV     @R1,@#$BDDAT    ;RHCS1 ---> $BDDAT
1784  010770  022737  104200  001126            CMP     #SC!DVA!RDY,@#$BDDAT ;COMPARE RHCS1 AFTER PARITY
1785                                                                    ;ERROR
1786  010776  001406                            BEQ     1$              ;BRANCH IF SC!DVA!RDY=1
1787  011000  012737  104200  001124            MOV     #SC!DVA!RDY,@#$GDDAT ;GOOD DATA
1788  011006  010137  042204                    MOV     R1,@#REGADR     ;REGISTER ADDRESS RHCS1
1789  011012  104001                            ERROR   1               ;SETTING PAT AND
1790                                                                    ;WRITING DCL REGISTER RHCS1
1791                                                                    ;DID NOT SET SC!DVA!RDY
1792                                                                    ;WITH 'PAT' BIT HIGH
1793  011014  013746  001774            1$:     MOV     @#UNIT,-(SP)    ;GET UNIT NUMBER
1794  011020  052716  000120                    BIS     #PAT!IR,(SP)    ;INCLUDE PAT AND IR
1795  011024  012637  001124                    MOV     (SP)+,@#$GDDAT  ;PUT ON STACK
1796  011030  011237  001126                    MOV     @R2,@#$BDDAT    ;RHCS2 ---> $BDDAT
1797  011034  023737  001124  001126            CMP     @#$GDDAT,@#$BDDAT ;COMPARE RHCS2
1798  011042  001403                            BEQ     2$              ;OK - SC!DVA!RDY ARE HIGH
1799  011044  010237  042204                    MOV     R2,@#REGADR     ;REGISTER ADDRESS
1800  011050  104001                            ERROR   1               ;READING DCL REGISTER RHCS2 DID NOT
1801                                                                    ;SHOW UNIT#!PAT!IR BITS HIGH
1802
1803  011052  011437  001126            2$:     MOV     @R4,@#$BDDAT    ;RHER1 ---> $BDDAT
1804  011056  022737  000010  001126            CMP     #PAR,@#$BDDAT   ;ERROR REGISTER RHER1 SHOULD
1805                                                                    ;HAVE 'PAR' SET
1806  011064  001406                            BEQ     3$              ;A - OK, IT DOES
1807  011066  012737  000010  001124            MOV     #PAR,@#$GDDAT   ;GOOD DATA
1808  011074  010437  042204                    MOV     R4,@#REGADR     ;FAILING REGISTER RHER1
1809  011100  104001                            ERROR   1               ;PARITY ERROR DID NOT
1810                                                                    ;SET 'PAR'
1811
1812  011102                            3$:
```

K4

```
1813
1814
1815
1816                                    ;;*******************************************************************
1817                                    ;*TEST 12        RHWC - WORD COUNT REGISTER
1818                                    ;**             TEST LOADING AND READING OF ALL POSSIBLE BITS IN THE HARDWARE
1819                                    ;**             REGISTERS.  USE A PATTERN OF WALKING 0'S (-2,-3,-5 ETC.) AND
1820                                    ;**             WALKING 1'S (1,2,4,10 ETC)
1821                                    ;;*******************************************************************
1822  011102  000004                   TST12:  SCOPE
1823
1824  011104  012706  001000                   MOV     #STACK,SP       ;RESET STACK
1825  011110  012737  000012  002032           MOV     #TTNO,@#TSTNM   ;THIS SAVES TEST NUMBER
1826
1827  011116  013737  001632  011140           MOV     @#RHWC,@#PRWC+14        ;GET REGISTER ADDRESS
1828  011124  013777  001774  170504  PRWC:    MOV     @#UNIT,@RHCS2   ;MOVE UNIT NO. UNDER TEST
1829  011132  004537  042206                   JSR     R5,BITST        ;TEST BITS IN REGISTER
1830  011136  177777                           .WORD   177777          ;ONLY THESE BITS ARE TESTED FOR READ/WRITE
1831  011140  176702                           .WORD   176702          ;ADDRESS OF REG. BEING TESTED
1832  011142  104001                           ERROR   1               ;INCORRECT DATA RECEIVED
1833  011144  000207                           RTS     PC              ;RETURN TO BLT3 ROUTINE
1834
1835
1836
1837
1838
1839
1840                                    ;;*******************************************************************
1841                                    ;*TEST 13        RHBA - UNIBUS ADDRESS REGISTER
1842                                    ;**             TEST LOADING AND READING OF ALL POSSIBLE BITS IN THE HARDWARE
1843                                    ;**             REGISTERS.  USE A PATTERN OF WALKING 0'S (-2,-3,-5 ETC.) AND
1844                                    ;**             WALKING 1'S (1,2,4,10 ETC)
1845                                    ;;*******************************************************************
1846  011146  000004                   TST13:  SCOPE
1847
1848  011150  012706  001000                   MOV     #STACK,SP       ;RESET STACK
1849  011154  012737  000013  002032           MOV     #TTNO,@#TSTNM   ;THIS SAVES TEST NUMBER
1850
1851  011162  013737  001634  011204           MOV     @#RHBA,@#PRBA+14        ;GET REGISTER ADDRESS
1852  011170  013777  001774  170440  PRBA:    MOV     @#UNIT,@RHCS2   ;MOVE UNIT NO. UNDER TEST
1853  011176  004537  042206                   JSR     R5,BITST        ;TEST BITS IN REGISTER
1854  011202  177776                           .WORD   177776          ;ONLY THESE BITS ARE TESTED FOR READ/WRITE
1855  011204  176704                           .WORD   176704          ;ADDRESS OF REG. BEING TESTED
1856  011206  104001                           ERROR   1               ;INCORRECT DATA RECEIVED
1857  011210  000207                           RTS     PC              ;RETURN TO BLT3 ROUTINE
1858
1859
1860
```

```
1861
1862
1863
1864                                    ;;*****************************************************************
1865                                    ;*TEST 14        RHER1 - ERROR REGISTER #1
1866                                    ;**             TEST LOADING AND READING OF ALL POSSIBLE BITS IN THE HARDWARE
1867                                    ;**             REGISTERS.  USE A PATTERN OF WALKING 0'S (-2,-3,-5 ETC.) AND
1868                                    ;**             WALKING 1'S (1,2,4,10 ETC)
1869                                    ;;*****************************************************************
1870  011212  000004                    TST14:  SCOPE
1871
1872  011214  012706  001000                    MOV     #STACK,SP       ;RESET STACK
1873  011220  012737  000014  002032            MOV     #TTNO,@#TSTNM   ;THIS SAVES TEST NUMBER
1874
1875  011226  013737  001642  011250            MOV     @#RHER1,@#PRER1+14      ;GET REGISTER ADDRESS
1876  011234  013777  001774  170374  PRER1:   MOV     @#UNIT,@RHCS2   ;MOVE UNIT NO. UNDER TEST
1877  011242  004537  042206                    JSR     R5,BITST        ;TEST BITS IN REGISTER
1878  011246  177777                            .WORD   177777          ;ONLY THESE BITS ARE TESTED FOR READ/WRITE
1879  011250  176714                            .WORD   176714          ;ADDRESS OF REG. BEING TESTED
1880  011252  104001                            ERROR   1               ;INCORRECT DATA RECEIVED
1881  011254  000207                            RTS     PC              ;RETURN TO BLT3 ROUTINE
1882
1883
```

```
1884
1885
1886
1887                                    ;;*****************************************************************
1888                                    ;*TEST 15       RHMR - MAINTENANCE REGISTER
1889                                    ;**      BIT 0 (DMD) MUST BE SET BEFORE THE OTHER BITS
1890                                    ;**      ARE READ WRITE
1891                                    ;**      ONLY 5 LOW ORDER BITS ARE TESTED (R2 HAS 5)
1892                                    ;;*****************************************************************
1893  011256  000004                   TST15:  SCOPE
1894
1895  011260  012737  000015  002032            MOV     #TTNO,@#TSTNM   ;THIS SAVES TEST NUMBER
1896  011266  004737  042534                    JSR     PC,@#CLDISK     ;SET UNIT NUMBER AND INIT
1897  011272  013700  001660                    MOV     @#RHMR, R0      ;R0 HAS MAINTENANCE REG. ADR.
1898  011276  012701  000001                    MOV     #1,R1           ;R1 HAS DATA
1899  011302  012702  000005                    MOV     #5,R2           ;R2 HAS COUNT OF NUMBER OF BITS
1900  011306  012710  000001           1$:      MOV     #DMD,@R0        ;SET DIAGNOSTIC MODE BIT
1901  011312  050110                            BIS     R1,@R0          ;SET DATA IN RHMR
1902  011314  010146                            MOV     R1,-(SP)        ;SAVE DATA FOR COMPARES
1903  011316  052716  000401                    BIS     #DMD!400,(SP)   ;INCLUDE BIT 0
1904  011322  011637  001124                    MOV     (SP),@#$GDDAT   ;SAVE FOR ERROR PRINTOUT
1905  011326  022610                            CMP     (SP)+,@R0       ;COMPARE DATA
1906  011330  001405                            BEQ     2$              ;BRANCH IF GOOD
1907  011332  011037  001126                    MOV     @R0,$BDDAT      ;BAD DATA
1908  011336  010037  042204                    MOV     R0,@#REGADR     ;FAILING REG. ADR.
1909  011342  104001                            ERROR   1               ;MAINTENANCE REGISTER
1910                                                                    ;FAILED TO SET INDICATED
1911                                                                    ;BITS
1912  011344  000241                   2$:      CLC                     ;CLEAR CARRY
1913  011346  006101                            ROL     R1              ;GET NEXT DATA
1914  011350  052701  000400                    BIS     #400,R1         ;SET UNUSED BITS
1915  011354  042701  001000                    BIC     #BIT09,R1       ;CLEAR READ ONLY BIT
1916  011360  005302                            DEC     R2              ;COUNT
1917  011362  001351                            BNE     1$              ;BRANCH IF 5 BITS NOT DONE
1918
1919
1920                                    ;*NOW FLOAT A 0
1921
1922  011364  012701  000435                    MOV     #435,R1         ;R1 HAS DATA
1923  011370  012702  000005                    MOV     #5,R2           ;R2 HAS COUNT BITS
1924  011374  012710  000001           3$:      MOV     #DMD,@R0        ;SET DIAGNOSTIC MODE BITS
1925  011400  050110                            BIS     R1,@R0          ;SET DATA IN RHMR
1926  011402  020110                            CMP     R1,@R0          ;COMPARE DATA
1927  011404  001407                            BEQ     4$              ;BRANCH IF GOOD
1928  011406  010137  001124                    MOV     R1,@#$GDDAT     ;GOOD DATA
1929  011412  011037  001126                    MOV     @R0,@#$BDDAT    ;BAD DATA
1930  011416  010037  042204                    MOV     R0,@#REGADR     ;FAILING REG. ADR. RHMR
1931  011422  104001                            ERROR   1               ;MAINTENANCE REGISTER
1932                                                                    ;DOES NOT ALLOW WRITING
1933                                                                    ;ZEROS
1934  011424  000261                   4$:      SEC                     ;SET CARRY
1935  011426  006101                            ROL     R1              ;GET NEXT DATA
1936  011430  042701  001340                    BIC     #BIT05!BIT06!BIT07!BIT09,R1  ;CLEAR READ ONLY BIT
1937  011434  052701  000400                    BIS     #BIT08,R1       ;SET BIT ZEROED BY ROL
1938  011440  005302                            DEC     R2              ;COUNT IF 5 BITS DONE
1939  011442  001354                            BNE     3$              ;BRANCH IF INCOMPLETE
```

```
1940
1941
```

```
1942
1943
1944
1945                                    ;;*****************************************************************
1946                                    ;*TEST 16        RHDST - DESIRED SECTOR/TRACK ADDRESS
1947                                    ;**             TEST LOADING AND READING OF ALL POSSIBLE BITS IN THE HARDWARE
1948                                    ;**             REGISTERS.  USE A PATTERN OF WALKING 0'S (-2,-3,-5 ETC.) AND
1949                                    ;**             WALKING 1'S (1,2,4,10 ETC)
1950                                    ;;*****************************************************************
1951  011444  000004                    TST16:  SCOPE
1952
1953
1954  011446  012706  001000                    MOV     #STACK,SP       ;RESET STACK
1955  011452  012737  000016  002032            MOV     #TTNO,@#TSTNM   ;THIS SAVES TEST NUMBER
1956
1957  011460  013737  001644  011502            MOV     @#RHDST,@#PRDST+14      ;GET REGISTER ADDRESS
1958  011466  013777  001774  170142  PRDST:    MOV     @#UNIT,@RHCS2   ;MOVE UNIT NO. UNDER TEST
1959  011474  004537  042206                    JSR     R5,BITST        ;TEST BITS IN REGISTER
1960  011500  017437                            .WORD   17437           ;ONLY THESE BITS ARE TESTED FOR READ/WRITE
1961  011502  176706                            .WORD   176706          ;ADDRESS OF REG. BEING TESTED
1962  011504  104001                            ERROR   1               ;INCORRECT DATA RECEIVED
1963  011506  000207                            RTS     PC              ;RETURN TO BLT3 ROUTINE
1964
1965
1966
1967
1968
1969
1970
1971                                    ;;*****************************************************************
1972                                    ;*TEST 17        RHER2 - ERROR REGISTER #2
1973                                    ;**             TEST LOADING AND READING OF ALL POSSIBLE BITS IN THE HARDWARE
1974                                    ;**             REGISTERS.  USE A PATTERN OF WALKING 0'S (-2,-3,-5 ETC.) AND
1975                                    ;**             WALKING 1'S (1,2,4,10 ETC)
1976                                    ;;*****************************************************************
1977  011510  000004                    TST17:  SCOPE
1978
1979
1980  011512  012706  001000                    MOV     #STACK,SP       ;RESET STACK
1981  011516  012737  000017  002032            MOV     #TTNO,@#TSTNM   ;THIS SAVES TEST NUMBER
1982
1983  011524  013737  001646  011546            MOV     @#RHER2,@#PRER2+14      ;GET REGISTER ADDRESS
1984  011532  013777  001774  170076  PRER2:    MOV     @#UNIT,@RHCS2   ;MOVE UNIT NO. UNDER TEST
1985  011540  004537  042206                    JSR     R5,BITST        ;TEST BITS IN REGISTER
1986  011544  177777                            .WORD   177777          ;ONLY THESE BITS ARE TESTED FOR READ/WRITE
1987  011546  176740                            .WORD   176740          ;ADDRESS OF REG. BEING TESTED
1988  011550  104001                            ERROR   1               ;INCORRECT DATA RECEIVED
1989  011552  000207                            RTS     PC              ;RETURN TO BLT3 ROUTINE
1990
1991
1992
1993
```

```
1994
1995
1996
1997                                    ;;*****************************************************************
1998                                    ;*TEST 20        RHOF - MARGIN/OFFSET REGISTER
1999                                    ;**             TEST LOADING AND READING OF ALL POSSIBLE BITS IN THE HARDWARE
2000                                    ;**             REGISTERS.  USE A PATTERN OF WALKING 0'S (-2,-3,-5 ETC.) AND
2001                                    ;**             WALKING 1'S (1,2,4,10 ETC)
2002                                    ;;*****************************************************************
2003  011554  000004                    TST20:  SCOPE
2004
2005  011556  012706  001000                    MOV     #STACK,SP       ;RESET STACK
2006  011562  012737  000020  002032            MOV     #TTNO,@#TSTNM   ;THIS SAVES TEST NUMBER
2007
2008  011570  013737  001650  011612            MOV     @#RHOF,@#PROF+14        ;GET REGISTER ADDRESS
2009  011576  013777  001774  170032  PROF:     MOV     @#UNIT,@RHCS2   ;MOVE UNIT NO. UNDER TEST
2010  011604  004537  042206                    JSR     R5,BITST        ;TEST BITS IN REGISTER
2011  011610  016277                            .WORD   16277           ;ONLY THESE BITS ARE TESTED FOR READ/WRITE
2012  011612  176732                            .WORD   176732          ;ADDRESS OF REG. BEING TESTED
2013  011614  104001                            ERROR   1               ;INCORRECT DATA RECEIVED
2014  011616  000207                            RTS     PC              ;RETURN TO BLT3 ROUTINE
2015
2016
2017
2018
2019
2020
2021
2022                                    ;;*****************************************************************
2023                                    ;*TEST 21        RHCA - DESIRED CYLINDER REGISTER
2024                                    ;**             TEST LOADING AND READING OF ALL POSSIBLE BITS IN THE HARDWARE
2025                                    ;**             REGISTERS.  USE A PATTERN OF WALKING 0'S (-2,-3,-5 ETC.) AND
2026                                    ;**             WALKING 1'S (1,2,4,10 ETC)
2027                                    ;;*****************************************************************
2028  011620  000004                    TST21:  SCOPE
2029
2030  011622  012706  001000                    MOV     #STACK,SP       ;RESET STACK
2031  011626  012737  000021  002032            MOV     #TTNO,@#TSTNM   ;THIS SAVES TEST NUMBER
2032
2033  011634  013737  001652  011656            MOV     @#RHCA,@#PRCA+14        ;GET REGISTER ADDRESS
2034  011642  013777  001774  167766  PRCA:     MOV     @#UNIT,@RHCS2   ;MOVE UNIT NO. UNDER TEST
2035  011650  004537  042206                    JSR     R5,BITST        ;TEST BITS IN REGISTER
2036  011654  001777                            .WORD   1777            ;ONLY THESE BITS ARE TESTED FOR READ/WRITE
2037  011656  176734                            .WORD   176734          ;ADDRESS OF REG. BEING TESTED
2038  011660  104001                            ERROR   1               ;INCORRECT DATA RECEIVED
2039  011662  000207                            RTS     PC              ;RETURN TO BLT3 ROUTINE
2040
```

```
2041
2042
2043                                    ;;*****************************************************************
2044                                    ;*TEST 22        RHER3 - ERROR REGISTER #3
2045                                    ;**             TEST LOADING AND READING OF ALL POSSIBLE BITS IN THE HARDWARE
2046                                    ;**             REGISTERS.  USE A PATTERN OF WALKING 0'S (-2,-3,-5 ETC.) AND
2047                                    ;**             WALKING 1'S (1,2,4,10 ETC)
2048                                    ;;*****************************************************************
2049  011664  000004                   TST22:  SCOPE
2050
2051  011666  012706  001000                   MOV     #STACK,SP       ;RESET STACK
2052  011672  012737  000022  002032           MOV     #TTNO,@#TSTNM   ;THIS SAVES TEST NUMBER
2053
2054  011700  013737  001654  011722           MOV     @#RHER3,@#PRER3+14      ;GET REGISTER ADDRESS
2055  011706  013777  001774  167722  PRER3:   MOV     @#UNIT,@RHCS2   ;MOVE UNIT NO. UNDER TEST
2056  011714  004537  042206                   JSR     R5,BITST        ;TEST BITS IN REGISTER
2057  011720  177777                           .WORD   177777          ;ONLY THESE BITS ARE TESTED FOR READ/WRITE
2058  011722  176742                           .WORD   176742          ;ADDRESS OF REG. BEING TESTED
2059  011724  104001                           ERROR   1               ;INCORRECT DATA RECEIVED
2060  011726  000207                           RTS     PC              ;RETURN TO BLT3 ROUTINE
2061
2062
2063
2064
2065
2066
2067                                    ;;*****************************************************************
2068                                    ;**OF THE TWENTY REGISTERS (4 IN RH11, 16 IN RP04) ONLY 12 ARE
2069                                    ;**CHECKED IN THE ABOVE TESTS
2070                                    ;**TWO ARE ALREADY TESTED (SERIAL NO. AND DRIVE TYPE)
2071                                    ;**THE OTHER 7 WHICH ARE RHDS1, RHLA, RHCC, RHEC1, RHEC1, RHEC2
2072                                    ;**ARE READ ONLY REGISTERS.  ONE OR ZERO CANNOT BE WRITTEN
2073                                    ;;*****************************************************************
2074
```

```
2075
2076
2077                                    ;;*****************************************************************
2078                                    ;*TEST 23       CONTROL AND STATUS 2 (RHCS 2) - 'NED'
2079                                    ;**     THIS TESTS THE UNIT SELECT BITS #0-2 (US1-4)
2080                                    ;**     AND NON-EXISTENT DRIVE BIT #12 (NED)
2081
2082                                    ;**     THE OTHER RHCS2 BITS ARE NOT TESTED HERE
2083
2084                                    ;;*****************************************************************
2085  011730  000004                    TST23:  SCOPE
2086  011732  012706  001000                    MOV     #STACK,SP       ;RESET STACK
2087  011736  012737  000023  002032            MOV     #TTNO,@#TSTNM   ;THIS SAVES TEST NUMBER
2088
2089  011744  004737  042534                    JSR     PC,@#CLDISK     ;HERE IT IS USED TO SETUP HARDWARE/
2090                                                                    ;CPU REGISTER CORRESPONDENCE
2091                                                                    ;R1=RHCS1
2092                                                                    ;R2=RHCS2
2093                                                                    ;R3=RHDS1
2094                                                                    ;R4=RHER1
2095  011750  005037  002006                    CLR     @#ERFLG$        ;CLEAR ERROR FLAG
2096
2097                                    ;*SIMULATED DISK AREA WILL BE USED AS A TEMPORARY
2098                                    ;*STORAGE TABLE FOR DRIVES PRESENT DETERMINED FROM 'NED' = 0 IN RHCS2
2099
2100  011754  012701  051116                    MOV     #DISK,R1        ;LOAD TABLE POINTER
2101  011760  012700  000010                    MOV     #8.,R0          ;LOAD TABLE LOCATION COUNTER
2102  011764  012721  177777            1$:     MOV     #-1,(R1)+       ;FILL 8 LOCATIONS WITH -1
2103  011770  005300                            DEC     R0              ;COUNT DOWN ONE LOCATION
2104  011772  001374                            BNE     1$              ;BRANCH IF 8 NOT DONE
2105
2106  011774  005012                            CLR     @R2             ;SELECT UNIT NO.0 (U2IU1IU0=0)
2107  011776  012700  000010                    MOV     #8.,R0          ;RELOAD TABLE LOCATION COUNTER
2108  012002  012701  051116                    MOV     #DISK,R1        ;RELOAD THE TABLE POINTER
2109  012006  005714                    2$:     TST     @R4             ;READ A DRIVE REGISTER (RHER1)
2110  012010  032712  010000                    BIT     #NED,@R2        ;NON EXISTENT DRIVE BIT = 0 ?
2111  012014  001415                            BEQ     3$              ;YES...DRIVE PRESENT, CHECK THE TYPE
2112  012016  005300                    7$:     DEC     R0              ;NO...DECREMENT DRIVE COUNT
2113  012020  001454                            BEQ     4$              ;CHECK RESULTS IF 8 DRIVES DONE
2114
2115  012022  011246                    10$:    MOV     @R2,-(SP)       ;PUT RHCS2 ON THE STACK
2116  012024  042716  177770                    BIC     #^C7,(SP)       ;MASK ALL BUT THE UNIT NUMBER
2117  012030  005216                            INC     (SP)            ;INCREMENT THE UNIT NUMBER
2118  012032  013703  001640                    MOV     @#RHCS1,R3      ;GET RHCS1 ADDRESS
2119  012036  005203                            INC     R3              ;ADDRESS UPPER BYTE OF RHCS1
2120  012040  112713  000100                    MOVB    #100,@R3        ;SET "TRE" IN RHCS1
2121                                                                    ;WITHOUT ADDRESSING DRIVE
2122  012044  012612                            MOV     (SP)+,@R2       ;PHCS2 HAS THE INCREMENTED UNIT
2123                                                                    ;WITH "NED" CLEARED
2124  012046  000757                            BR      2$              ;TEST FOR NEXT DRIVE
2125
2126
2127
2128                                    ;*CHECK THE UNIT TYPE AND BUILD "NED" DERIVED UNITS TABLE
2129
2130  012050  022777  024020  167606     3$:     CMP     #24020,@RHDT    ;IS THIS A DUAL PORT RP04 ?
```

F5

```
2131  012056  001425                    BEQ     8$              ;ENTER IN TABLE IF SO
2132  012060  022777  020020  167576    CMP     #20020,@RHDT    ;IS THIS A SINGLE PORT RP04 ?
2133  012066  001421                    BEQ     8$              ;ENTER IN TABLE IF SO
2134
2135                                ;;**************************************************************
2136  012070  022777  024022  167566    CMP     #24022,@RHDT    ;IS THIS A DUAL PORT RP06 ?
2137  012076  001415                    BEQ     8$              ;ENTER IN TABLE IF SO
2138  012100  022777  020022  167556    CMP     #20022,@RHDT    ;IS THIS A SINGLE PORT RP06 ?
2139  012106  001411                    BEQ     8$              ;ENTER IN TABLE IF SO
2140
2141  012110  022777  024021  167546    CMP     #24021,@RHDT    ;IS THIS A DUAL PORT RP05 ?
2142  012116  001405                    BEQ     8$              ;ENTER IN TABLE IF SO
2143  012120  022777  020021  167536    CMP     #20021,@RHDT    ;IS THIS A SINGLE PORT RP05 ?
2144  012126  001401                    BEQ     8$              ;ENTER IN TABLE IF SO
2145                                ;;**************************************************************
2146
2147  012130  000732                    BR      7$              ;NO RP04 FOUND SO CHECK NEXT UNIT
2148
2149  012132  012746  000010    8$:     MOV     #8.,-(SP)       ;LOAD MAX NO. OF DRIVES
2150  012136  160016                    SUB     R0,(SP)         ;(SP) NOW HAS THE PRESENT DRIVE NO.
2151  012140  012621                    MOV     (SP)+,(R1)+     ;LOAD TABLE, INCR TABLE LOCATION &
2152                                                            ;RESTORE THE STACK TO WHERE IT WAS
2153  012142  005300                    DEC     R0              ;DECREMENT THE DRIVE COUNT
2154  012144  001402                    BEQ     4$              ;CHECK RESULTS IF 8 UNITS CHECKED
2155  012146  005212                    INC     @R2             ;SELECT NEXT UNIT
2156  012150  000716                    BR      2$              ;GO TEST IT
2157
2158
2159                                ;*COMPARE 'NED' DERIVED UNITS TABLE WITH THAT DERIVED USING RHAS IN T4
2160
2161  012152  004037  043430    4$:     JSR     R0,@#COMPAR     ;COMPARE RESULTS
2162  012156  001754                    UNITS                   ;RHER1/RHAS DERIVED DATA
2163  012160  051116                    DISK                    ;'NED' TEST DATA
2164  012162  000010                    8.                      ;NO.OF WORDS TO COMPARE
2165  012164  012172                    5$                      ;RETURN FOR ERROR HEADER
2166  012166  012220                    6$                      ;RETURN FOR ERROR DATA
2167  012170  012336                    13$                     ;RETURN FOR GOOD COMPARISON (NEXT TEST)
2168
2169
2170
2171                                ;*SPECIAL 'NED'/'RHAS' TABLE TYPE OUT ROUTINE (BYPASSES .SERRTYP AND
2172                                ;*HENCE IGNORES INHIBIT ERROR TYPEOUT SWITCH)
2173
2174  012172  104022    5$:     ERROR   22
2175  012174  012703  000010            MOV     #8.,R3          ;LENGTH OF BOTH UNIT TABLES
2176  012200  012701  001754            MOV     #UNITS,R1       ;ADDRESS OF RHAS/RHER1 UNITS TABLE
2177  012204  012702  051116            MOV     #DISK,R2        ;ADDRESS OF 'NED' RHCS2 UNITS TABLE
2178  012210  012137  001124    14$:    MOV     (R1)+,@#$GDDAT  ;LOAD RHAS UNIT NO.INTO "$GDDAT" AND
2179                                                            ;INCREMENT THE TABLE LOCATION
2180  012214  012237  001126            MOV     (R2)+,@#$BDDAT  ;LOAD 'NED' UNIT NO. INTO "$BDDAT"
2181                                                            ;& INCR TABLE LOCATION
2182
2183  012220  032777  020000  166712  6$:     BIT     #SW13,@SWR      ;INHIBIT ERROR TYPE OUTS ?
2184  012226  001043                    BNE     13$             ;YES...EXIT
2185  012230  022737  177777  001124    CMP     #-1,@#$GDDAT    ;DOES RHAS UNIT TABLE LOCATION = -1 ?
2186  012236  001413                    BEQ     11$             ;YES...DON'T TYPE IT - CHECK 'NED' TABLE
```

```
012240  104401  062417                  TYPE    ,SPACE8         ;NO...TAB OVER PC COLUMN
012244  104401  062417                  TYPE    ,SPACE8         ;TAB OVER THE TEST NO. COLUMN
012250  013746  001124                  MOV     @#$GDDAT,-(SP)  ;;SAVE @#$GDDAT FOR TYPEOUT
012254  104403                          TYPOS                   ;;GO TYPE--OCTAL ASCII
012256     006                          .BYTE   6               ;;TYPE 6 DIGITS
012257     000                          .BYTE   0               ;;SUPPRESS LEADING ZEROS
012260  104401  062425                  TYPE    ,SPACE2         ;SPACE OVER TO THE NEXT COLUMN
012264  000406                          BR      12$             ;CHECK THE 'NED' UNIT TABLE

012266  104401  062417          11$:    TYPE    ,SPACE8         ;TAB OVER THE PC COMUMN
012272  104401  062417                  TYPE    ,SPACE8         ;TAB OVER THE TEST NO. COLUMN
012276  104401  062417                  TYPE    ,SPACE8         ;TAB OVER THE RHAS UNIT COLUMN

012302  022737  177777  001126  12$:    CMP     #-1,@#$BDDAT    ;DOES 'NED' UNIT TABLE LOCATION = - 1 ?
012310  001404                          BEQ     9$              ;YES...DON'T TYPE IT
012312  013746  001126                  MOV     @#$BDDAT,-(SP)  ;;SAVE @#$BDDAT FOR TYPEOUT
012316  104403                          TYPOS                   ;;GO TYPE--OCTAL ASCII
012320     006                          .BYTE   6               ;;TYPE 6 DIGITS
012321     000                          .BYTE   0               ;;SUPPRESS LEADING ZEROS

012322  104401  001223          9$:     TYPE    ,$CRLF          ;FOR THE NEXT LINE IN BOTH TABLES
012326  005303                          DEC     R3              ;COUNT DOWN 2 TABLES LOCATION COUNTER
012330  001327                          BNE     14$             ;IF NOT = 0 TYPE OUT NEXT 2 LOCATIONS
012332  062706  000014                  ADD     #14,SP          ;ADJUST STACK FOR NO "POP" & RTS FROM "COMPAR"

012336                          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


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

H5

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

I5

```
2246
2247
2248
2249
2250                                    ;;*****************************************************************
2251                                    ;*TEST 24       CONTROL AND STATUS 2 (RHCS2) - 'CLR'
2252                                    ;*THIS TESTS THE UNIT SELECT BITS (US1-4) AND CLEAR BIT #5 (CLR)
2253
2254                                    ;*ALL REGISTERS ARE LOADED WITH ALL ONES EXCEPT BIT #0 AND #6
2255                                    ;*WHICH ARE "GO" AND "INTERRUPT ENABLE", THEN 'CLR' IS GIVEN
2256                                    ;*(RHDB IS READ FIRST AS THIS WILL SET DTL IN RHCS2 AND
2257                                    ;*SC AND TRE IN RHCS1.)
2258
2259                                    ;*ANOTHER CLR IS GIVEN THEN ALL OTHER REGISTERS ARE READ
2260                                    ;;*****************************************************************
2261  012336  000004                    TST24:  SCOPE
2262
2263  012340  004737  042534                    JSR     PC,@#CLDISK     ;SET REGISTERS AND CLEAR
2264  012344  005037  002006                    CLR     @#ERFLG$        ;CLEAR ANY ERRORS
2265
2266                                            ;*FILL ALL POSSIBLE BITS WITH ONES
2267
2268  012350  012777  177777  167252          MOV     #177777,@RHDB   ;BUS ADDRESS REGISTER GETS 177777
2269  012356  012777  177777  167246          MOV     #177777,@RHWC   ;WORD COUNT REGISTER GETS 177777
2270  012364  012777  177777  167242          MOV     #177777,@RHBA   ;BUS ADDRESS REGISTER GETS 177777
2271  012372  052777  157010  167236          BIS     #157010,@RHCS2  ;CONTROL AND STATUS 2 GETS 157010
2272  012400  012777  001476  167232          MOV     #1476,@RHCS1    ;CONTROL AND STATUS REGISTER/GETS 1476
2273  012406  012777  177777  167226          MOV     #177777,@RHER1  ;ERROR REGISTER1 GETS 177777
2274  012414  012777  017437  167222          MOV     #17437,@RHDST   ;DESIRED SECTOR TRACK
2275  012422  012777  177777  167216          MOV     #177777,@RHER2  ;ERROR REGISTER 2
2276  012430  012777  016277  167212          MOV     #16277,@RHOF    ;OFFSET REGISTER
2277  012436  012777  177777  167206          MOV     #177777,@RHCA   ;DESIRED CYLINDER
2278  012444  012777  177777  167202          MOV     #177777,@RHER3  ;ERROR REGISTER 3
2279  012452  012777  000001  167200          MOV     #DMD,@RHMR      ;MAINTENANCE REGISTER
2280  012460  012777  177777  167172          MOV     #177777,@RHMR   ;MAINTENANCE REGISTER
2281
2282  012466  052712  000040                  BIS     #CLR,   @R2     ;CLEAR ALL POSSIBLE BITS
2283  012472  013712  001774                  MOV     @#UNIT,@R2      ;REINSTATE UNIT NO.
2284  012476  012700  001630                  MOV     #RHDB,  R0      ;R0 CONTAINS ADDR. OF ADDR. OF REG.
2285
2286                                            ;*DATA BUFFER REGISTER
2287
2288  012502  012737  177777  001124          MOV     #177777,@#$GDDAT;GOOD DATA FOR ERROR
2289  012510  011037  042204                  MOV     @R0,@#REGADR    ;REGISTER ADDRESS
2290  012514  013037  001126                  MOV     @(R0)+,@#$BDDAT ;TEST DATA
2291  012520  023737  001124  001126          CMP     @#$GDDAT,@#$BDDAT;COMPARE GOOD WITH TEST DATA
2292  012526  001401                          BEQ     2$              ;BRANCH IF GOOD
2293  012530  104001                          ERROR   1               ;RHDB DID NOT HAVE ALL ONES
2294                                                                  ;AFTER A CLR IN RHCS2
2295
2296  012532  052712  000040          2$:     BIS     #CLR,@R2        ;SET CLEAR AGAIN BECAUSE
2297                                                                  ;READING RHDB AFTER CLEARING  WILL
2298                                                                  ;SET DLT SC AND TRE
2299  012536  013712  001774                  MOV     @#UNIT,@R2      ;REINSTATE UNIT NO.
2300
2301                                            ;*WORD COUNT REGISTER
```

```
2302
2303
2304  012542  012737  177777  001124  3$:   MOV   #177777,@#$GDDAT        ;GOOD DATA FOR ERROR TYPEOUT
2305  012550  013037  001126              MOV   @(R0)+,@#$BDDAT ;TEST DATA
2306  012554  022737  177777  001126      CMP   #177777,@#$BDDAT        ;COMPARE DATA
2307  012562  001402                      BEQ   4$                ;BRANCH IF GOOD
2308  012564  004737  013350              JSR   PC,@#ERCS2C       ;JUMP TO ERROR FOR CLR (BIT 5)
2309                                                              ;IN RHCS2
2310
2311                                      ;*BUS ADDRESS REGISTER
2312
2313
2314  012570  012737  000000  001124  4$:   MOV   #0,@#$GDDAT       ;GOOD DATA FOR ERROR TYPEOUT
2315  012576  013037  001126              MOV   @(R0)+,@#$BDDAT ;TEST DATA
2316  012602  022737  000000  001126      CMP   #0,@#$BDDAT       ;COMPARE DATA
2317  012610  001402                      BEQ   5$                ;BRANCH IF GOOD
2318  012612  004737  013350              JSR   PC,@#ERCS2C       ;JUMP TO ERROR FOR CLR (BIT 5)
2319                                                              ;IN RHCS2
2320
2321                                      ;*CONTROL AND STATUS 2 REGISTER
2322
2323
2324  012616  012746  000100        5$:   MOV   #100,-(SP)        ;INCLUDE IR
2325  012622  053716  001774              BIS   @#UNIT,(SP)       ;SET UNIT NO.
2326  012626  012637  001124              MOV   (SP)+,@#$GDDAT    ;GOOD DATA FOR TYPE OUT
2327  012632  013037  001126              MOV   @(R0)+,@#$BDDAT ;TEST DATA
2328  012636  023737  001124  001126      CMP   @#$GDDAT,@#$BDDAT;COMPARE DATA
2329  012644  001402                      BEQ   6$                ;BRANCH IF GOOD
2330  012646  004737  013350              JSR   PC,@#ERCS2C       ;JUMP TO ERROR FOR CLR (BIT 5)
2331                                                              ;IN RHCS2
2332
2333
2334                                      ;*CONTROL AND STATUS 1 REGISTER
2335
2336
2337  012652  012737  004276  001124  6$:   MOV   #4276,@#$GDDAT    ;GOOD DATA FOR ERROR TYPEOUT
2338  012660  013037  001126              MOV   @(R0)+,@#$BDDAT ;TEST DATA
2339  012664  022737  004276  001126      CMP   #4276,@#$BDDAT    ;COMPARE DATA
2340  012672  001402                      BEQ   7$                ;BRANCH IF GOOD
2341  012674  004737  013350              JSR   PC,@#ERCS2C       ;JUMP TO ERROR FOR CLR (BIT 5)
2342                                                              ;IN RHCS2
2343
2344                                      ;*ERROR 1 REGISTER
2345
2346
2347  012700  012737  000000  001124  7$:   MOV   #0,@#$GDDAT       ;GOOD DATA FOR ERROR TYPEOUT
2348  012706  013037  001126              MOV   @(R0)+,@#$BDDAT ;TEST DATA
2349  012712  022737  000000  001126      CMP   #0,@#$BDDAT       ;COMPARE DATA
2350  012720  001402                      BEQ   10$               ;BRANCH IF GOOD
2351  012722  004737  013350              JSR   PC,@#ERCS2C       ;JUMP TO ERROR FOR CLR (BIT 5)
2352                                                              ;IN RHCS2
2353
2354                                      ;*DESIRED SECTOR/TRACK REGISTER
2355
2356
2357  012726  012737  017437  001124  10$:   MOV   #17437,@#$GDDAT        ;GOOD DATA FOR ERRON TYPEOUT
```

K5

```
2358  012734  013037  001126                    MOV     @(R0)+,@#$BDDAT ;TEST DATA
2359  012740  022737  017437  001126            CMP     #17437,@#$BDDAT ;COMPARE DATA
2360  012746  001402                            BEQ     11$             ;BRANCH IF GOOD
2361  012750  004737  013350                    JSR     PC,@#ERCS2C     ;JUMP TO ERROR FOR CLR (BIT 5)
2362                                                                    ;IN RHCS2
2363
2364                                            ;*ERROR 2 REGISTER
2365
2366
2367  012754  012737  000000  001124  11$:      MOV     #0,@#$GDDAT     ;GOOD DATA FOR ERROR TYPEOUT
2368  012762  013037  001126                    MOV     @(R0)+,@#$BDDAT ;TEST DATA
2369  012766  022737  000000  001126            CMP     #0,@#$BDDAT     ;COMPARE DATA
2370  012774  001402                            BEQ     12$             ;BRANCH IF GOOD
2371  012776  004737  013350                    JSR     PC,@#ERCS2C     ;JUMP TO ERROR FOR CLR (BIT 5)
2372                                                                    ;IN RHCS2
2373
2374                                            ;*OFFSET REGISTER
2375
2376
2377  013002  012737  116000  001124  12$:      MOV     #116000,@#$GDDAT        ;GOOD DATA FOR ERROR TYPEOUT
2378  013010  013037  001126                    MOV     @(R0)+,@#$BDDAT ;TEST DATA
2379  013014  022737  116000  001126            CMP     #116000,@#$BDDAT        ;COMPARE DATA
2380  013022  001402                            BEQ     13$             ;BRANCH IF GOOD
2381  013024  004737  013350                    JSR     PC,@#ERCS2C     ;JUMP TO ERROR FOR CLR (BIT 5)
2382                                                                    ;IN RHCS2
2383
2384                                            ;*DESIRED CYLINDER ADDRESS REGISTER
2385
2386
2387  013030  012737  001777  001124  13$:      MOV     #1777,@#$GDDAT  ;GOOD DATA FOR ERROR TYPEOUT
2388  013036  013037  001126                    MOV     @(R0)+,@#$BDDAT ;TEST DATA
2389  013042  022737  001777  001126            CMP     #1777,@#$BDDAT  ;COMPARE DATA
2390  013050  001402                            BEQ     14$             ;BRANCH IF GOOD
2391  013052  004737  013350                    JSR     PC,@#ERCS2C     ;JUMP TO ERROR FOR CLR (BIT 5)
2392                                                                    ;IN RHCS2
2393
2394                                            ;*ERROR 3 REGISTER
2395
2396
2397  013056  012737  000000  001124  14$:      MOV     #0,@#$GDDAT     ;GOOD DATA FOR ERROR TYPEOUT
2398  013064  013037  001126                    MOV     @(R0)+,@#$BDDAT ;TEST DATA
2399  013070  022737  000000  001126            CMP     #0,@#$BDDAT     ;COMPARE DATA
2400  013076  001402                            BEQ     15$             ;BRANCH IF GOOD
2401  013100  004737  013350                    JSR     PC,@#ERCS2C     ;JUMP TO ERROR FOR CLR (BIT 5)
2402                                                                    ;IN RHCS2
2403
2404                                            ;*ATTENTION SUMMARY REGISTER
2405
2406  013104  013037  001126            15$:    MOV     @(R0)+,@#$BDDAT         ;GET RHAS CONTENTS
2407  013110  012737  000000  001124            MOV     #0,@#$GDDAT             ;GOOD DATA FOR ERROR TYPE OUT
2408  013116  123737  001124  001126            CMPB    @#$GDDAT,@#$BDDAT       ;COMPARE FOR RHAS
2409  013124  001402                            BEQ     16$             ;BRANCH IF GOOD
2410  013126  004737  013350                    JSR     PC,@#ERCS2C             ;JUMP TO ERROR FOR CLR (BIT 5)
2411                                                                    ;IN RHCS2
2412
2413                                            ;*MAINTAINABILITY REGISTER
```

```
2414
2415
2416  013132  012737  000400  001124  16$:  MOV   #400,@#$GDDAT   ;GOOD DATA FOR ERROR TYPEOUT
2417  013140  013037  001126              MOV   @(R0)+,@#$BDDAT ;TEST DATA
2418  013144  022737  000400  001126      CMP   #400,@#$BDDAT   ;COMPARE DATA
2419  013152  001402                      BEQ   17$             ;BRANCH IF GOOD
2420  013154  004737  013350              JSR   PC,@#ERCS2C     ;JUMP TO ERROR FOR CLR (BIT 5)
2421                                                            ;IN RHCS2
2422
2423                                      ;*DRIVE STATUS REGISTER
2424
2425  013160  012737  000600  001124  17$:  MOV   #600,@#$GDDAT   ;GOOD DATA FOR ERROR TYPEOUT
2426  013166  013046                      MOV   @(R0)+,-(SP)    ;GET RHDS1
2427  013170  011637  001126              MOV   (SP),@#$BDDAT   ;TEST DATA
2428  013174  042716  001100              BIC   #VV!PROG,(SP)   ;CLEAR VV AND PROG
2429  013200  022726  000600              CMP   #600,(SP)+      ;COMPARE DATA
2430  013204  001402                      BEQ   20$             ;BRANCH IF GOOD
2431  013206  004737  013350              JSR   PC,@#ERCS2C     ;JUMP TO ERROR FOR CLR (BIT 5)
2432                                                            ;IN RHCS2
2433
2434                                      ;*DRIVE TYPE
2435
2436
2437  013212  013737  002010  001124  20$:  MOV   @#SAVDT,@#$GDDAT       ;GOOD DATA FOR ERROR TYPEOUT
2438  013220  013037  001126              MOV   @(R0)+,@#$BDDAT ;TEST DATA
2439  013224  023737  002010  001126      CMP   @#SAVDT,@#$BDDAT       ;COMPARE DATA
2440  013232  001402                      BEQ   21$             ;BRANCH IF GOOD
2441  013234  004737  013350              JSR   PC,@#ERCS2C     ;JUMP TO ERROR FOR CLR (BIT 5)
2442                                                            ;IN RHCS2
2443
2444
2445                                      ;*SERIAL NUMBER REGISTER
2446
2447
2448
2449  013240  013737  002012  001124  21$:  MOV   @#SAVSN,@#$GDDAT       ;GOOD DATA FOR ERROR TYPEOUT
2450  013246  013037  001126              MOV   @(R0)+,@#$BDDAT ;TEST DATA
2451  013252  023737  002012  001126      CMP   @#SAVSN,@#$BDDAT       ;COMPARE DATA
2452  013260  001402                      BEQ   22$             ;BRANCH IF GOOD
2453  013262  004737  013350              JSR   PC,@#ERCS2C     ;JUMP TO ERROR FOR CLR (BIT 5)
2454                                                            ;IN RHCS2
2455
2456
2457                                      ;*ECC1 POSITION
2458
2459
2460  013266  012737  000000  001124  22$:  MOV   #0,@#$GDDAT     ;GOOD DATA FOR ERROR TYPEOUT
2461  013274  013037  001126              MOV   @(R0)+,@#$BDDAT ;TEST DATA
2462  013300  022737  000000  001126      CMP   #0,@#$BDDAT     ;COMPARE DATA
2463  013306  001402                      BEQ   23$             ;BRANCH IF GOOD
2464  013310  004737  013350              JSR   PC,@#ERCS2C     ;JUMP TO ERROR FOR CLR (BIT 5)
2465                                                            ;IN RHCS2
2466
2467
2468                                      ;*ECC2 PATTERN
2469
```

MS

```
2470
2471  013314  012737  000000  001124  23$:    MOV     #0,@#$GDDAT     ;GOOD DATA FOR ERROR TYPEOUT
2472  013322  013037  001126          MOV     @(R0)+,@#$BDDAT ;TEST DATA
2473  013326  022737  000000  001126  CMP     #0,@#$BDDAT     ;COMPARE DATA
2474  013334  001402                  BEQ     24$             ;BRANCH IF GOOD
2475  013336  004737  013350          JSR     PC,@#ERCS2C     ;JUMP TO ERROR FOR CLR (BIT 5)
2476                                                          ;IN RHCS2
2477
2478
2479                                  ;*LOOK-AHEAD REGISTER
2480
2481  013342  005720          24$:    TST     (R0)+           ;AS THE LOOK-AHEAD REG. CANNOT BE PREDICTED
2482                                                          ;AFTER AN INIT IT IS NOT CHECKED
2483
2484                                  ;*CURRENT CYLINDER ADDRESS REGISTER
2485
2486  013344  005720          25$:    TST     (R0)+           ;AS THE CURRENT REG. CANNOT BE PREDICTED
2487                                                          ;AFTER A INIT IT IS NOT CHECKED
2488
2489  013346                  26$:
2490  013346  000405                  BR      TST25           ;BRANCH OVER JSR
2491
2492
2493  013350  014037  042204  ERCS2C: MOV     -(R0),  @#REGADR ;FAILING REGISTER ADDRESS
2494  013354  104001                  ERROR   1               ;CLR (BIT 5) IN RHCS2 DID
2495                                                          ;NOT CLEAR APPROPIATE BITS
2496                                                          ;OR CLEARED EXTRA BITS
2497  013356  005720                  TST     (R0)+           ;UNDO -(R0) FOR BAD DATA
2498  013360  000207                  RTS     PC              ;RETURN TO TEST ABOVE
```

```
2499
2500
2501                                    ;;*****************************************************************
2502                                    ;*TEST 25       PACK ACKNOWLEDGE COMMAND TEST
2503
2504                                    ;**     THE PACK ACKNOWLEDGE COMMAND WILL BE LOADED INTO RHCS1 WITH GO
2505                                    ;**     THEN ALL REGISTERS WILL BE CHECKED
2506                                    ;**     RH CLEAR WILL BE GIVEN
2507                                    ;**     THEN ALL REGISTERS WILL BE CHECKED
2508
2509                                    ;;*****************************************************************
2510  013362  000004                    TST25:  SCOPE
2511  013364  012706  001000                    MOV     #STACK,SP       ;RESET STACK
2512  013370  012737  000025  002032            MOV     #TTNO,@#TSTNM   ;THIS SAVES TEST NUMBER
2513
2514  013376  004737  042534                    JSR     PC,@#CLDISK     ;INIT AND SET UP GENERAL CPU/DEVICE
2515                                                                    ;REGISTER CORRESPONDENCE AND UNIT NO.
2516  013402  012777  000001  166250            MOV     #DMD,@RHMR      ;SET DIAGNOSTIC MODE
2517  013410  013777  002100  166222            MOV     @#PKACK,@RHCS1  ;LOAD "PACK ACKNOWLEDGE COMMAND" INTO RHCS1
2518
2519                                            ;*SAVE REGISTERS FOR COMPARISON AFTER 'GO' IS ISSUED
2520  013416  004037  043226                    JSR     R0,@#SAVER      ;SAVE
2521  013422  001632                            RHWC                    ;FROM
2522  013424  003154                            REINTO                  ;TO
2523  013426  000023                            19.                     ;NUMBER OF REGISTERS SAVED
2524
2525
2526  013430  052777  000001  166202            BIS     #GO,@RHCS1      ;ISSUE 'GO' TO PACK ACKNOWLEDGE COMMAND
2527
2528                                            ;*CHANGE SAVED REGISTERS TO EXPECTED VALUES
2529  013436  052737  000100  003204            BIS     #VV,@#REINTO+30 ;SAVED RHDS1
2530
2531                                            ;*AFTER GO HAS BEEN GIVEN TO PACK ACKNOWLEDGE COMMAND
2532                                            ;*SAVE REGISTERS AGAIN SO THAT COMPARISONS CAN
2533                                            ;*BE DONE
2534
2535  013444  004037  043226                    JSR     R0,@#SAVEP      ;SAVE
2536  013450  001632                            RHWC                    ;FROM
2537  013452  002110                            WRFROM
2538  013454  000023                            19.                     ;NUMBER OF REGISTERS SAVED
2539
2540                                            ;*AS UPPER BYTE OF RHAS CAN BE CHANGING IN A DUAL PORT
2541                                            ;*OPERATION THE UPPER BYTE OF RHAS WILL BE SAVED AS IS
2542                                            ;*SO THAT THE COMPARES ARE ONLY VALID FOR THE LOWER BYTE
2543  013456  113737  003201  002135            MOVB    @#REINTO+25,@#WRFROM+25;SAVE UPPER RHAS
2544
2545
2546                                            ;*COMPARE REGISTERS BEFORE PACK ACKNOWLEDGE COMMAND
2547                                            ;*WITH AFTER GO
2548
2549  013464  004037  043430                    JSR     R0,@#COMPAR     ;COMPARE
2550  013470  003154                            REINTO                  ;GOOD BUFFER
2551  013472  002110                            WRFROM                  ;TEST BUFFER
2552  013474  000023                            19.                     ;NUMBER
2553  013476  013504                            1$                      ;RETURN FOR ERROR
2554  013500  013504                            1$                      ;SAME
```

```
2555  013502  013524                          2$                      ;RETURN FOR GOOD COMPARISON
2556
2557  013504  013705  047320          1$:     MOV     @#ERWORD,R5     ;GETTING READY TO INDEX
2558  013510  060505                          ADD     R5,R5           ;DOUBLE ERROR WORD
2559  013512  016537  001630  042204          MOV     RHWC-2(R5),@#REGADR ;FAILING REGISTER ADDRESS
2560
2561  013520  104001                          ERROR   1               ;IMPROPER REGISTER CHANGE
2562                                                                  ;AFTER PACK ACKNOWLEDGE COMMAND
2563                                                                  ;WITH GO IS GIVEN
2564  013522  000207                          RTS     PC              ;RETURN TO COMPARISON
2565
2566  013524                          2$:                             ;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
                                ;;*****************************************************************
013524  000004                  TST26:  SCOPE
013526  012706  001000                  MOV     #STACK,SP       ;RESET STACK
013532  012737  000026  002032          MOV     #TTNO,@#TSTNM   ;THIS SAVES TEST NUMBER
013540  004737  042534                  JSR     PC,@#CLDISK     ;INIT AND SET UP GENERAL CPU/DEVICE
                                                                ;REGISTER CORRESPONDENCE

                                        ;*FILL ALL POSSIBLE REGISTER BITS WITH ONES

013544  012777  177777  166060          MOV     #177777,@RHWC   ;WORD COUNT REGISTER GETS 177777
013552  012777  177777  166054          MOV     #177777,@RHBA   ;BUS ADDRESS REGISTER GETS 177777
013560  052777  157010  166050          BIS     #157010,@RHCS2  ;CONTROL AND STATUS 2 GETS 177430
013566  012777  001476  166044          MOV     #1476,@RHCS1    ;CONTROL AND STATUS REGISTER 1 GETS 21476
013574  012777  177777  166040          MOV     #177777,@RHER1  ;ERROR REGISTER1 GETS 177777
013602  012777  017437  166034          MOV     #17437,@RHDST   ;DESIRED SECTOR TRACK
013610  012777  177777  166030          MOV     #177777,@RHER2  ;ERROR REGISTER 2
013616  012777  016277  166024          MOV     #16277,@RHOF    ;OFFSET REGISTER
013624  012777  000777  166020          MOV     #777,@RHCA      ;DESIRED CYLINDER
013632  012777  177777  166014          MOV     #177777,@RHER3  ;ERROR REGISTER 3
013640  012777  000001  166012          MOV     #DMD,@RHMR      ;MAINTENANCE REGISTER
013646  012777  177777  166004          MOV     #177777,@RHMR   ;MAINTENANCE REGISTER

                                        ;*BEFORE RESET SAVE REGISTERS IN READ INTO BUFFER
013654  004037  043226                  JSR     R0,@#SAVER      ;SAVE
013660  001632                          RHWC                    ;FROM
013662  003154                          REINTO                  ;TO
013664  000021                          17.                     ;NUMBER

                                        ;*GIVE RESET AND REINSTATE UNIT NUMBER
013666  000005                          RESET
013670  004737  054672                  JSR     PC,@#$TKINT     ;INITIALIZE TK
013674  053777  001774  165734          BIS     @#UNIT,@RHCS2

                                        ;*CHANGE ORIGINAL SAVED REGISTERS TO EXPECTED VALUES AFTER RESET
013702  005037  003156                  CLR     @#REINTO+2      ;CLEAR SAVED RHBA
013706  013746  001774                  MOV     @#UNIT,-(SP)    ;GET UNIT NUMBER FRO SAVED RHCS2
013712  052716  000100                  BIS     #IR,(SP)        ;INCLUDE IR
013716  012637  003160                  MOV     (SP)+,@#REINTO+4 ;SAVED RHCS2
013722  012737  004276  003162          MOV     #DVAIRDY!76,@#REINTO+6 ;SAVED RHCS1
013730  005037  003164                  CLR     @#REINTO+10     ;SAVED RHER1
013734  005037  003170                  CLR     @#REINTO+14     ;SAVED RHER2
013740  012737  116000  003172          MOV     #116000,@#REINTO+16 ;SAVED RHOF
013746  005037  003176                  CLR     @#REINTO+22     ;SAVED RHER3
013752  105037  003200                  CLRB    @#REINTO+24     ;SAVED RHAS
013756  012737  000400  003202          MOV     #400,@#REINTO+26;SAVED RHMR
```

DL

```
                                        ;*CHANGE RHDS1 WITHOUT CHANGING PROG BIT
013764  013746  003204                  MOV     @#REINTO+30,-(SP) ;GET RHDS1
013770  042716  176777                  BIC     #^CPROG,(SP)    ;CLEAR EVERYTHING EXCEPT PROG
013774  052716  000700                  BIS     #700,(SP)       ;SET EXPECTED BITS - 'DPR', 'DRY' & 'VV'

014000  012637  003204          4$:     MOV     (SP)+,@#REINTO+30;SAVED RHDS1
014004  005037  003212                  CLR     @#REINTO+36     ;SAVED RHEC1
014010  005037  003214                  CLR     @#REINTO+40     ;SAVED RHEC2

                                        ;*AFTER RESET, SAVE REGISTERS FOR COMPARISONS TO BE DONE
014014  004037  043226                  JSR     R0,@#SAVER      ;SAVE
014020  001632                          RHWC                    ;FROM
014022  002110                          WRFROM                  ;TO
014024  000021                          17.                     ;NUMBER

                                        ;*AS UPPER BYTE OF RHAS CAN BE CHANGING IN A DUAL PORT
                                        ;*OPERATION THE UPPER BYTE OF RHAS WILL BE SAVED AS IS
                                        ;*SO THAT THE COMPARES ARE ONLY VALID FOR THE LOWER BYTE
014026  113737  003201  002135          MOVB    @#REINTO+25,@#WRFROM+25;SAVE UPPER RHAS

                                        ;*COMPARE REGISTERS BEFORE RESET WITH REGISTERS AFTER RESET
014034  004037  043430                  JSR     R0,@#COMPAR     ;COMPARE
014040  003154                          REINTO                  ;GOOD BUFFER
014042  002110                          WRFROM                  ;TEST BUFFER
014044  000021                          17.                     ;NUMBER
014046  014054                          1$                      ;RETURN FOR ERROR
014050  014054                          1$                      ;SAME
014052  014074                          2$                      ;RETURN FOR GOOD COMPARISON

014054  013705  047320          1$:     MOV     @#ERWORD,R5     ;GETTING READY TO INDEX
014060  060505                          ADD     R5,R5           ;DOUBLE ERROR WORD
014062  016537  001630  042204          MOV     RHWC-2(R5),@#REGADR ;FAILING REGISTER ADDRESS
014070  104001                          ERROR   1               ;REGISTER CONTENTS AFTER
                                                                ;A RESET THAT IS AN
                                                                ;UNIBUS INITIALIZE CAUSED
                                                                ;AN IMPROPER REGISTER CHANGE
014072  000207                          RTS     PC              ;RETURN TO COMPARISON
014074                          2$:                             ;RETURN TO POINT ON GOOD COMPARISON
```

```
2661
2662                                    .SBTTL  SILO TESTS
2663
2664
2665                                ;;**********************************************************************
2666                                ;*TEST 27        SILO TST 1
2667
2668                                ;**     THIS TESTS THE SILO BUFFER IN THE RH11 CONTROLLER
2669                                ;**     A READ IS ATTEMPTED FROM AN EMPTY SILO
2670                                ;**     DATA LATE (DLT) (RHCS2), TRANSFER ERROR (TRE) (RHCS1),
2671                                ;**     SPECIAL CONDITION (SC) (RHCS1) SHOULD SET
2672                                ;**     THEN LOADING "1" INTO TRE SHOULD CLEAR DLT, TRE AND SC
2673
2674                                ;;**********************************************************************
2675  014074  000004                TST27:  SCOPE
2676  014076  012706  001000               MOV     #STACK,SP       ;RESET STACK
2677  014102  012737  000027  002032       MOV     #TTNO,@#TSTNM   ;THIS SAVES TEST NUMBER
2678
2679                                ;*CHECK TO SEE IF THE PROGRAM IS RUNNING WITH AN RH70
2680
2681  014110  005737  002040               TST     @#RH70          ;TEST FLAG FOR RH70 CONTROLLER
2682  014114  001402                       BEQ     30$             ;IF FLAG = 1, THIS TEST IS SKIPPED
2683  014116  000137  014252               JMP     TST30           ;JUMP TO NEXT TEST ------------------)
2684  014122                        30$:                            ;IF FLAG = 0, DO THIS TEST
2685
2686  014122  004737  042534               JSR     PC,CLDISK       ;CLEAR DISK AND LOAD R'S
2687
2688  014126  017700  165476               MOV     @RHDB,R0        ;READ FROM EMPTY SILO
2689  014132  013746  001774               MOV     @#UNIT,-(SP)    ;GET UNIT NO. IN
2690  014136  052716  100100               BIS     #DLT!IR,(SP)    ;GET DATA LATE BIT AND IR
2691  014142  004737  042140               JSR     PC,@#PUTREG     ;SAVE REGISTERS
2692  014146  022637  001712               CMP     (SP)+,@#CS2     ;IS DATA LATE BIT UP?
2693  014152  001403                       BEQ     1$              ;IF YES BRANCH
2694  014154  010237  001122               MOV     R2,@#$BDADR     ;IF NOT STORE FAILING REG.
2695  014160  104011                       ERROR   11              ;RHCS2 DID NOT HAVE DLT
2696                                                               ;RHCS2 SHOULD HAVE ONLY
2697                                                               ;DLT AND UNIT NUMBER (BIT 0-2)
2698                                                               ;ALL OTHER BITS SHOULD
2699                                                               ;BE 0
2700  014162  022737  144200  001714 1$:     CMP     #SC!TRE!RDY!DVA,@#CS1 ;IS SPECIAL CONDITION, TRANSFER ERROR
2701                                                               ;AND READY UP?
2702  014170  001403                       BEQ     2$              ;IF YES BRANCH
2703  014172  010137  001122               MOV     R1,@#$BDADR     ;IF NOT STORE FAILING REG.
2704  014176  104011                       ERROR   11              ;RHCS1 DID NOT HAVE SC, DVA
2705                                                               ;TRE AND RDY. AFTER A
2706                                                               ;READ FROM EMPTY SILO ONLY
2707                                                               ;THESE BITS SHOULD BE UP
2708                                                               ;ALL OTHERS SHOULD BE 0
2709  014200  012711  040000        2$:     MOV     #TRE,@R1        ;CLEAR ERROR BITS BY MOVING
2710                                                               ;ONE INTO TRE IN RHCS1
2711  014204  004737  042140               JSR     PC,@#PUTREG     ;SAVE REGISTERS
2712  014210  022737  004200  001714       CMP     #RDY!DVA,@#CS1  ;ALL BITS BUT RDY AND DVA SHOULD
2713                                                               ;BE 0
2714  014216  001403                       BEQ     3$              ;BRANCH IF YES
2715  014220  010137  001122               MOV     R1,@#$BDADR     ;STORE FAILING ADDRESS
2716  014224  104011                       ERROR   11              ;AFTER A ONE IN TRE ONLY
```

```
2717                                                                ;READY AND DVA SHOULD BE SET IN
2718                                                                ;RHCS1
2719  014226  013746  001774        3$:   MOV     @#UNIT,-(SP)
2720  014232  052716  000100              BIS     #IR,(SP)
2721  014236  022637  001712              CMP     (SP)+,@#CS2       ;RHCS2 SHOULD HAVE IR AND UNIT ONLY
2722  014242  001403                      BEQ     TST30             ;BRANCH IF YES
2723  014244  010237  001122              MOV     R2,@#SBDADR       ;STORE FAILING ADDR
2724  014250  104011                      ERROR   11                ;AFTER A ONE IN TRE ONLY
```

```
2725
2726
2727                                    ;;*****************************************************************
2728                                    ;*TEST 30       SILO TEST 2
2729
2730                                    ;**     THIS TESTS THE IR AND "OR" BITS OF RHCS2
2731                                    ;**     AT THE BEGINNING IR SHOULD BE SET AND "OR" RESET
2732                                    ;**     LOADING 0 IN SILO RESETS IR FOR ONLY 2 MICRO SECONDS
2733                                    ;**     THIS TIME CANNOT BE CHECKED BUT IT IS CHECKED TO SEE IF
2734                                    ;**     IT DOES GO DOWN OR NOT
2735                                    ;**     THEN ALL 1 IS LOADED IN SILO "OR" SHOULD BECOME SET
2736                                    ;**     IN 30 MICRO SECONDS AGAIN TIME IS NOT  CHECKED
2737                                    ;**     "OR" SHOULD BE SET
2738                                    ;**     THE OUTPUT FROM THE SILO SHOULD BE 0 AND ALL ONES
2739
2740                                    ;;*****************************************************************
2741  014252  000004                    TST30:  SCOPE
2742  014254  012706  001000                    MOV     #STACK,SP       ;RESET STACK
2743  014260  012737  000030  002032            MOV     #TTNO,@#TSTNM   ;THIS SAVES TEST NUMBER
2744
2745                                            ;*CHECK TO SEE IF THE PROGRAM IS RUNNING WITH AN RH70
2746
2747  014266  005737  002040                    TST     @#RH70          ;TEST FLAG FOR RH70 CONTROLLER
2748  014272  001402                            BEQ     30$             ;IF FLAG = 1, THIS TEST IS SKIPPED
2749  014274  000137  014500                    JMP     TST31           ;JUMP TO NEXT TEST ----------------)
2750  014300                            30$:                            ;IF FLAG = 0, DO THIS TEST
2751
2752  014300  004737  042534                    JSR     PC,CLDISK       ;CLEAR REGISTERS LOAD R'S
2753
2754  014304  013746  001774                    MOV     @#UNIT,-(SP)
2755  014310  052716  000100                    BIS     #IR,(SP)
2756  014314  004737  042140                    JSR     PC,@#PUTREG     ;SAVE REGISTERS
2757  014320  022637  001712                    CMP     (SP)+,@#CS2     ;IR SHOULD BE SET "OR" RESET
2759  014324  001403                            BEQ     1$
2759  014326  010237  001122                    MOV     R2,@#SBDADR     ;FAILING REGISTER RHCS2
2760  014332  104011                            ERROR   11              ;RHCS2 DOES NOT HAVE IR
2761                                                                    ;SET, UNIT NO. SET AND
2762                                                                    ;ALL OTHER BITS 0
2763  014334  005077  165270           1$:     CLR     @RHDB           ;LOAD DATA BUFFER (SILO) WITH 0
2764  014340  012777  177777  165262            MOV     #-1,@RHDB       ;LOAD SILO WITH ALL ONES
2765  014346  013737  001636  014356            MOV     @#RHCS2,@#2$    ;ADDRESS OF RHCS2
2766  014354  104415                            WAT                     ;WAIT TRAP
2767  014356  000000                    2$:     .WORD                   ;ADDRESS OF RHCS2
2768  014360  000200                            OR
2769  014362  013746  001774           3$:     MOV     @#UNIT,-(SP)    ;
2770  014366  052716  000300                    BIS     #OR!IR,(SP)     ;IR AND "OR"
2771  014372  004737  042140                    JSR     PC,@#PUTREG     ;SAVE REGISTERS
2772  014376  022637  001712                    CMP     (SP)+,@#CS2     ;IR AND "OR" SHOULD BE SET
2773  014402  001403                            BEQ     4$
2774  014404  010237  001122                    MOV     R2,@#SBDADR     ;SAVE RHCS2 ADDR. FAILING REG.
2775  014410  104011                            ERROR   11              ;"OR" IN RHCS2 SHOULD BE
2776                                                                    ;SET TOGETHER WITH IR AND
2777                                                                    ;UNIT NO.
2778  014412  017700  165212           4$:     MOV     @RHDB,R0        ;SAVE SILO DATA SHOULD BE 0
2779  014416  017705  165206                    MOV     @RHDB,R5        ;SAVE SILO DATA SHOULD BE ALL 1
2780  014422  022700  000000                    CMP     #0,R0           ;FIRST WORD 0? XYZ DO MORE TEST
```

H6

```
2781  014426  001410                          BEQ     5$                ;BRANCH IF YES
2782  014430  005037  001124                  CLR     @#$GDDAT          ;GOOD DATA
2783
2784  014434  010037  001126                  MOV     R0,@#$BDDAT       ;BAD DATA
2785  014440  013737  001630  042204          MOV     @#RHDB,@#REGADR   ;SAVE RHDB FAILING REG.
2786  014446  104001                          ERROR   1                 ;SILO DID NOT HAVE THE FIRST WORD
2787                                                                    ;"0" WHEN "OR" WAS SET
2788  014450  022705  177777          5$:     CMP     #-1,R5            ;SECOND WORD ALL ONES?
2789  014454  001411                          BEQ     TST31             ;BRANCH IF YES
2790  014456  012737  177777  001124          MOV     #-1,@#$GDDAT      ;GOOD DATA
2791  014464  010537  001126                  MOV     R5,@#$BDDAT       ;BAD DATA
2792  014470  013737  001630  042204          MOV     @#RHDB,@#REGADR   ;SAVE RHDB FAILING REG.
2793  014476  104001                          ERROR   1                 ;SILO DID NOT HAVE THE SECOND
2794                                                                    ;WORD OF ALL ONES WHEN "OR"
2795                                                                    ;WAS SET
2796
```

```
2797
2798
2799                                    ;;*****************************************************************
2800                                    ;*TEST 31        SILO TEST 3
2801
2802                                    ;**     THIS TESTS SILO BUFFER BY FILLING IT WITH A COUNT FROM
2803                                    ;**     0 TO 65 AND THEN CHECKING IF IR IS DOWN AND "OR"
2804                                    ;**     IS HIGH AND COMPARING THE SILO OUTPUT.
2805
2806                                    ;;*****************************************************************
2807  014500  000004                    TST31:  SCOPE
2808  014502  012737  000031  002032            MOV     #TTNO,@#TSTNM   ;THIS SAVES TEST NUMBER
2809
2810                                            ;*CHECK TO SEE IF THE PROGRAM IS RUNNING WITH AN RH70
2811
2812  014510  005737  002040                    TST     @#RH70          ;TEST FLAG FOR RH70 CONTROLLER
2813  014514  001402                            BEQ     30$             ;IF FLAG = 1, THIS TEST IS SKIPPED
2814  014516  000137  014740                    JMP     TST32           ;JUMP TO NEXT TEST ----------------------)
2815  014522                            30$:                            ;IF FLAG = 0, DO THIS TEST
2816
2817
2818  014522  012700  051116                    MOV     #SILOTB,R0      ;TABLE POINTER
2819  014526  012705  000103                    MOV     #67.,R5         ;COUNTER
2820  014532  005020                    1$:     CLR     (R0)+           ;CLEAR TOTAL TABLE
2821  014534  005305                            DEC     R5              ;COUNT
2822  014536  001375                            BNE     1$              ;BRANCH IF NOT COMPLETELY CLEAR
2823  014540  004737  042534                    JSR     PC,@#CLDISK     ;CLEAR ALL REG.
2824  014544  005000                            CLR     R0
2825  014546  012705  000102                    MOV     #66.,R5         ;COUNT
2826  014552  010077  165052            2$:     MOV     R0,@RHDB        ;LOAD SILO WITH COUNT FROM 0 TO 65
2827  014556  005200                            INC     R0              ;NEXT COUNT
2828  014560  005305                            DEC     R5              ;IS 66 LOADS DONE?
2829  014562  001373                            BNE     2$              ;BRANCH IF NOT.
2830  014564  013746  001774                    MOV     @#UNIT,-(SP)
2831  014570  052716  000200                    BIS     #OR,(SP)
2832  014574  004737  042140                    JSR     PC,@#PUTREG     ;SAVE REGISTERS
2833  014600  022637  001712                    CMP     (SP)+,@#CS2     ;"OR" SHOULD BE SET IR RESET
2834  014604  001405                            BEQ     3$              ;BRANCH IF YES
2835  014606  010237  001122                    MOV     R2,@#$BDADR     ;SAVE RHCS2 ADR. FAILING REG.
2836  014612  104011                            ERROR   11              ;"OR" WAS NOT SET, IR WAS NOT
2837  014614  005037  002006                    CLR     @#ERFLG$        ;RESET AFTER SILO WAS FULL
2838  014620  012700  051116            3$:     MOV     #SILOTB,R0      ;POINTER
2839  014624  012705  000102                    MOV     #66.,R5         ;COUNTER
2840  014630  017720  164774            4$:     MOV     @RHDB,(R0)+     ;READ SILO
2841  014634  005305                            DEC     R5              ;COUNT
2842  014636  001374                            BNE     4$              ;BRANCH IF 66 NOT DONE
2843  014640  012700  051116                    MOV     #SILOTB,R0      ;POINTER
2844  014644  012705  000102                    MOV     #66.,R5
2845  014650  005046                            CLR     -(SP)
2846  014652  021620                    5$:     CMP     (SP),(R0)+
2847  014654  001425                            BEQ     7$              ;BRANCH IF GOOD
2848  014656  014037  001126                    MOV     -(R0),@#$BDDAT  ;BAD DATA
2849  014662  011637  001124                    MOV     (SP),@#$GDDAT   ;GOOD DATA
2850  014666  013737  001630  042204          MOV     @#RHDB,@#REGADR ;FAILING REG. RHDB
2851  014674  005737  002006                    TST     @#ERFLG$        ;IS THIS FIRST ERROR?
2852  014700  001002                            BNE     6$              ;IF NOT BRANCH
```

```
2853  014702  104012                    ERROR   12              ;THESE TWO ERROR CALLS ARE FOR
2854  014704  000401                    BR      64$             ;BRANCH TO AVOID PRINTING NEXT ERROR
2855  014706  104013            6$:     ERROR   13              ;THE SAME TYPEOUT.  SILO
2856                                                            ;HAD A COUNT WRITTEN IN.
2857                                                            ;ON READ OUT AN ERROR WAS
2858                                                            ;DETECTED. THE TOTAL SILO
2859                                                            ;READOUT IS IN LOCATION
2860                                                            ;"SILOTB" TO THE NEXT 65
2861                                                            ;WORDS.
2862  014710  005720            64$:    TST     (R0)+           ;INCREMENT (R0)
2863                                                            ;ARE FURTHER COMPARES TO
2864  014712  017746  164222            MOV     @SWR,-(SP)      ;BE DONE
2865  014716  042716  177577            BIC     #^CSW07!SW08,(SP) ;ONLY KEEP SW7 AND SW8
2866  014722  022726  000200            CMP     #SW07,(SP)+     ;TEST SW07
2867  014726  001403                    BEQ     10$             ;IF NO MORE COMPARE THEN BRANCH
2868  014730  005216            7$:     INC     (SP)            ;NEXT GOOD WORD
2869  014732  005305                    DEC     R5              ;COUNT
2870  014734  001346                    BNE     5$              ;BRANCH IF 66 NOT COMPLETE
2871  014736  005726            10$:    TST     (SP)+           ;POP STACK
```

```
2872
2873
2874
2875                                    ;;*******************************************************************
2876                                    ;*TEST 32        SILO TEST4
2877                                    ;**     NOW PUT 67 WORDS INTO SILO AND CHECK FOR DLT
2878                                    ;**     EVEN AFTER THE 67TH. WORD INPUT THE FIRST WORD SHOULD NOT CHANGE
2879
2880                                    ;;*******************************************************************
2881  014740  000004                    TST32:  SCOPE
2882  014742  012737  000032  002032            MOV     #TTNO,@#TSTNM   ;THIS SAVES TEST NUMBER
2883
2884                                            ;*CHECK TO SEE IF THE PROGRAM IS RUNNING WITH AN RH70
2885
2886  014750  005737  002040                    TST     @#RH70          ;TEST FLAG FOR RH70 CONTROLLER
2887  014754  001402                            BEQ     30$             ;IF FLAG = 1, THIS TEST IS SKIPPED
2888  014756  000137  015064                    JMP     TST33           ;JUMP TO NEXT TEST ----------------------)
2889  014762                            30$:                            ;IF FLAG = 0, DO THIS TEST
2890
2891  014762  004737  042534                    JSR     PC,@#CLDISK     ;CLEAR DISK REG.
2892
2893  014766  005000                            CLR     R0              ;CLEAR R0
2894  014770  005200                    1$:     INC     R0              ;ADD 1
2895  014772  010077  164632                    MOV     R0,@RHDB        ;LOAD SILO
2896  014776  022700  000103                    CMP     #67.,R0         ;67 DONE?
2897  015002  001401                            BEQ     2$              ;BRANCH IF YES
2898  015004  000771                            BR      1$              ;NO SO BRANCH
2899  015006  004737  042140            2$:     JSR     PC,@#PUTREG     ;SAVE REGISTERS
2900
2901  015012  032737  100000  001712            BIT     #DLT,@#CS2      ;DLT SET?
2902  015020  001003                            BNE     3$              ;BRANCH IF YES
2903  015022  010237  001122                    MOV     R2,@#$BDADR     ;FAILING ADDRESS RHCS2
2904  015026  104011                            ERROR   11              ;DATA LATE DID NOT SET AT 67TH.
2905  015030  017737  164574  001126    3$:     MOV     @RHDB,@#$BDDAT  ;INPUT TO SILO
2906  015036  012737  000001  001124            MOV     #1,@#$GDDAT     ;GOOD DATA
2907  015044  023737  001124  001126            CMP     @#$GDDAT,@#$BDDAT ;COMPARE
2908  015052  001404                            BEQ     TST33           ;BRANCH IF GOOD
2909  015054  013737  001630  042204            MOV     @#RHDB,@#REGADR ;FAILING REG. RHDB
2910  015062  104012                            ERROR   12              ;WORD IN RHDB CHANGED
2911                                            ;AFTER THE 67TH INPUT.
```

```
                                ;;*****************************************************************
                                ;*TEST 33       SILO TEST 5
                                ;**     THE SILO IS LOADED WITH 0,1,2,3 THEN AFTER
                                ;**     "OR" IS UP A CLR IN RHCS2 IS DONE THEN 4,
                                ;**     IS LOADED. AFTER "OR" IS UP 2 READS FROM
                                ;**     SILO ARE DONE, ON THE LAST, "DTL" IN RHCS2 SHOULD BE SET.
                                ;;*****************************************************************
015064  000004          TST33:  SCOPE

015066  012737  000033  002032          MOV     #TTNO,@#TSTNM   ;THIS SAVES TEST NUMBER

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

015074  005737  002040          TST     @#RH70          ;TEST FLAG FOR RH70 CONTROLLER
015100  001402                  BEQ     30$             ;IF FLAG = 1, THIS TEST IS SKIPPED
015102  000137  015364          JMP     TST34           ;JUMP TO NEXT TEST ------------------)
015106                  30$:                            ;IF FLAG = 0, DO THIS TEST

015106  004737  042534          JSR     PC,@#CLDISK     ;CLEAR DISK

015112  013746  001774          MOV     @#UNIT,-(SP)    ;GET UNIT NO.
015116  052716  000100          BIS     #IR,(SP)        ;SET INPUT READY
015122  004737  042140          JSR     PC,@#PUTREG     ;SAVE REGISTERS
015126  022637  001712          CMP     (SP)+,@#CS2     ;IR SHOULD BE SET "OR" CLEARED
015132  001403                  BEQ     1$              ;BRANCH IF GOOD
015134  010237  001122          MOV     R2,@#$BDADR     ;FAILING REGISTER RHCS2
015140  104011                  ERROR   11              ;RHCS2 DOES NOT HAVE IR SET
                                                        ;AND ALL OTHER BITS 0
015142  013700  001630  1$:     MOV     @#RHDB,R0       ;R0 HAS RHDB ADDRESS
015146  005001                  CLR     R1              ;DATA
015150  010110          2$:     MOV     R1,@R0          ;0, THEN 1 THEN 2 THEN 3
                                                        ;IN RHDB
015152  005201                  INC     R1              ;INCREMENT DATA
015154  022701  000004          CMP     #4,R1           ;IS 4 DONE
015160  103373                  BHIS    2$              ;BRANCH IF NOT
015162  013737  001636  015172  MOV     @#RHCS2,@#3$    ;
015170  104415                  WAT                     ;WAIT FOR "OR"
015172  000000          3$:     .WORD   0               ;RHCS2 ADDRESS
015174  000200                  OR                      ;WAIT ON OR.
015176  004737  042534          JSR     PC,@#CLDISK     ;CLR IN RHCS2
015202  013746  001774          MOV     @#UNIT,-(SP)    ;UNIT NO.
015206  052716  000100          BIS     #IR,(SP)
015212  004737  042140          JSR     PC,@#PUTREG     ;SAVE REGISTERS
015216  022637  001712          CMP     (SP)+,@#CS2     ;IR SHOULD BE SET "0"=0
015222  001403                  BEQ     4$              ;BRANCH IF GOOD
015224  010237  001122          MOV     R2,@#$BDADR     ;FAILING REGISTER RHCS2
015230  104011                  ERROR   11              ;RHCS2 DOES NOT HAVE IR SET
                                                        ;AND ALL OTHER BITS 0
015232  013700  001630  4$:     MOV     @#RHDB,R0       ;R0 HAS RHDB ADDRESS
015236  012710  000004          MOV     #4,@R0          ;LOAD 4 IN SILO
015242  011201                  MOV     @R2,R1          ;SAVE RHCS2
015244  011005                  MOV     @R0,R5          ;READ THE 4 IN SILO
```

M6

```
2968  015246  011003                        MOV     @R0,R3          ;READ SILO TO GET DLT
2969  015250  011204                        MOV     @R2,R4          ;SAVE RHCS2
2970  015252  032701  000200                BIT     #OR,R1          ;TEST FOR OR IN RHCS2
2971  015256  001424                        BEQ     6$              ;IF OR IS NOT SET BRANCH
2972  015260  022705  000004                CMP     #4,R5           ;SILO 4 IS NOW COMPARED
2973  015264  001410                        BEQ     5$
2974  015266  010037  042204                MOV     R0,@#REGADR     ;SILO ADDRESS
2975  015272  012737  000004  001124        MOV     #4,@#$GDDAT     ;GOOD DATA
2976  015300  010537  001126                MOV     R5,@#$BDDAT     ;BAD DATA
2977  015304  104001                        ERROR   1               ;SILO DID NOT CONTAIN WORD
2978                                                                ;PUT IN AFTER "OR" WAS UP
2979  015306  005703                5$:     TST     R3              ;IS IT ZERO BECAUSE SILO
2980                                                                ;IS DESTRUCTIVE READ
2981  015310  001407                        BEQ     6$              ;BRANCH IF GOOD
2982  015312  010037  042204                MOV     R0,@#REGADR     ;SILO ADDRESS
2983  015316  005037  001124                CLR     @#$GDDAT        ;GOOD DATA
2984  015322  010337  001126                MOV     R3,@#$BDDAT     ;BAD DATA
2985  015326  104001                        ERROR   1               ;SILO SHOULD BE ZERO
2986                                                                ;AFTER THE ONE WORD PUT IN
2987                                                                ;HAS BEEN TAKEN OUT AS
2988                                                                ;SILO IS A DESTRUCTIVE READ
2989  015330  032704  100000        6$:     BIT     #DLT,R4         ;
2990  015334  001013                        BNE     TST34           ;BRANCH IF DLT SET
2991  015336  013746  001774                MOV     @#UNIT,-(SP)    ;GET UNIT NO
2992  015342  052716  100300                BIS     #DLT!OR!IR,(SP) ;
2993  015346  012637  001124                MOV     (SP)+,@#$GDDAT  ;GOOD DATA
2994  015352  010437  001126                MOV     R4,@#$BDDAT     ;BAD DATA
2995  015356  010237  042204                MOV     R2,@#REGADR     ;RHCS2 ADDRESS
2996  015362  104001                        ERROR   1               ;DATA LATE ERROR
```

```
                                .SBTTL  MORE REGISTER TESTS

                                ;;*********************************************************************
                                ;*TEST 34       TEST ODD BYTE INSTRUCTION ON RHCS1 - RH11

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

                                ;;*********************************************************************
015364  000004                  TST34:  SCOPE

015366  012737  000034  002032          MOV     #TTNO,@#TSTNM   ;THIS SAVES TEST NUMBER

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

015374  005737  002040                  TST     @#RH70          ;TEST FLAG FOR RH70 CONTROLLER
015400  001402                          BEQ     30$             ;IF FLAG = 1, THIS TEST IS SKIPPED
015402  000137  015522                  JMP     TST35           ;JUMP TO NEXT TEST ----------------------)
015406                          30$:                            ;IF FLAG = 0, DO THIS TEST

015406  012706  001000                  MOV     #STACK,SP       ;RESET STACK
015412  004737  042534                  JSR     PC,CLDISK       ;CLEAR DISK REG.

015416  012711  003566                  MOV     #3566,@R1       ;LOAD RHCS1 WITH ANY NUMBER
015422  010146                          MOV     R1,-(SP)        ;GETTING READY TO FORM ODD BYTE
015424  005216                          INC     (SP)            ;SP NOW HAS ODD BYTE FOR RHCS1
015426  112736  000005                  MOVB    #5,@(SP)+       ;MOVE 5 INTO ODD BYTE FOR RHCS1
015432  011137  001126                  MOV     @R1,@#$BDDAT    ;TEST DATA
015436  022737  006766  001126          CMP     #2566!DVA!RDY,@#$BDDAT ;RHCS1 SHOULD HAVE 6766
015444  001406                          BEQ     1$              ;BRANCH IF GOOD
015446  012737  006766  001124          MOV     #2566!DVA!RDY,@#$GDDAT ;GOOD DATA
FAIFAHHIHHNG£NG @E @GISGIPDEPDR RR @@C@@S1 S1

015460  104001                          ERROR   1               ;MOVING A NUMBER INTO
                                                                ;ODD BYTE OF RHCS1 GAVE
                                                                ;WRONG RESULTS

015462  112711  000032          1$:     MOVB    #32,@R1         ;MOVE INTO EVEN BYTE
015466  011137  001126                  MOV     @R1,@#$BDDAT    ;TEST DATA
015472  022737  006632  001126          CMP     #2432!DVA!RDY,@#$BDDAT ;RHCS1 SHOULD HAVE 6632
015500  001460                          BEQ     TST36           ;DO NEXT RH11 TEST IF GOOD ---------------)
015502  012737  006632  001124          MOV     #2432!DVA!RDY,@#$GDDAT ;GOOD DATA
015510  010137  042204                  MOV     R1,@#REGADR     ;FAILING REGISTER RHCS1
015514  104001                          ERROR   1               ;MOVING A NUMBER INTO EVEN
                                                                ;BYTE OF RHCS1 GAVE WRONG
                                                                ;RESULT

015516  000137  015642                  JMP     TST36           ;SKIP RH70 TEST ----------------------)
```

```
3047
3048
3049                                    ;;*******************************************************************
3050                                    ;*TEST 35       TEST ODD BYTE INSTRUCTION ON RHCS1 - RH70
3051
3052                                    ;**     RDY (BIT 07) AND DVA (BIT 11) SHOULD ALWAYS BE SET
3053
3054                                    ;;*******************************************************************
3055  015522  000004                    TST35:  SCOPE
3056
3057  015524  012737  000035  002032            MOV     #TTNO,@#TSTNM   ;THIS SAVES TEST NUMBER
3058  015532  012706  001000                    MOV     #STACK,SP       ;RESET STACK
3059  015536  004737  042534                    JSR     PC,CLDISK       ;CLEAR DISK REG.
3060
3061  015542  012711  003566                    MOV     #3566,@R1       ;LOAD RHCS1 WITH ANY NUMBER
3062  015546  010146                            MOV     R1,-(SP)        ;GETTING READY TO FORM ODD BYTE
3063  015550  005216                            INC     (SP)            ;SP NOW HAS ODD BYTE FOR RHCS1
3064  015552  112736  000005                    MOVB    #5,@(SP)+       ;MOVE 5 INTO ODD BYTE FOR RHCS1
3065  015556  011137  001126                    MOV     @R1,@#$BDDAT    ;TEST DATA
3066
3067  015562  022737  004766  001126            CMP     #566!DVA!RDY,@#$BDDAT ;RHCS1 SHOULD HAVE 4766
3068  015570  001406                            BEQ     1$              ;BRANCH IF GOOD
3069  015572  012737  004766  001124            MOV     #566!DVA!RDY,@#$GDDAT ;GOOD DATA
3070  015600  010137  042204                    MOV     R1,@#REGADR     ;FAILING REGISTER RHCS1
3071  015604  104001                            ERROR   1               ;MOVING A NUMBER INTO
3072                                                                    ;ODD BYTE OF RHCS1 GAVE
3073                                                                    ;WRONG RESULTS
3074
3075  015606  112711  000032            1$:     MOVB    #32,@R1         ;MOVE INTO EVEN BYTE
3076  015612  011137  001126                    MOV     @R1,@#$BDDAT    ;TEST DATA
3077  015616  022737  004632  001126            CMP     #432!DVA!RDY,@#$BDDAT ;RHCS1 SHOULD HAVE 4632
3078  015624  001406                            BEQ     TST36           ;;BRANCH IF GOOD
3079  015626  012737  004632  001124            MOV     #432!DVA!RDY,@#$GDDAT ;GOOD DATA
3080  015634  010137  042204                    MOV     R1,@#REGADR     ;FAILING REGISTER RHCS1
3081  015640  104001                            ERROR   1               ;MOVING A NUMBER INTO EVEN
3082                                                                    ;BYTE OF RHCS1 GAVE WRONG
3083                                                                    ;RESULTS
3084
```

```
3085
3086
3087                                    ;;*****************************************************************
3088                                    ;*TEST 36       TEST ODD BYTE INSTRUCTION ON RHCS2
3089
3090                                    ;**      IR (BIT 06) AND THE UNIT SELECT (BIT 0-2) WILL BE SET
3091
3092                                    ;;*****************************************************************
3093  015642  000004                    TST36:  SCOPE
3094
3095  015644  012737  000036  002032            MOV     #TTNO,@#TSTNM   ;THIS SAVES TEST NUMBER
3096
3097                                    ;*CHECK TO SEE IF THE PROGRAM IS RUNNING WITH AN RH70
3098
3099  015652  005737  002040            TST     @#RH70          ;TEST FLAG FOR RH70 CONTROLLER
3100  015656  001402                    BEQ     30$             ;IF FLAG = 1, THIS TEST IS SKIPPED
3101  015660  000137  016020            JMP     TST37           ;JUMP TO NEXT TEST ------------------)
3102  015664                    30$:                            ;IF FLAG = 0, DO THIS TEST
3103
3104  015664  004737  042534            JSR     PC,@#CLDISK     ;GIVE INIT & SETUP REGISTER CORRES
3105
3106  015670  052712  177000            BIS     #177000,(P2)    ;LOAD RHCS2
3107  015674  010246                    MOV     R2,-(SP)        ;GETTING READY FOR ODD BYTE
3108  015676  005216                    INC     (SP)            ;SP NOW HAS ODD BYTE FOR RHCS2
3109  015700  105036                    CLRB    @(SP)+          ;CLRR RHCS2 ODD BYTE
3110  015702  013746  001774            MOV     @#UNIT, -(SP)   ;GET UNIT NO.
3111  015706  052716  000100            BIS     #IR,(SP)        ;INPUT READY AS IT IS SET
3112  015712  011237  001126            MOV     @R2,@#$BDDAT    ;TEST DATA
3113  015716  022637  001126            CMP     (SP)+,@#$BDDAT  ;COMPARE TO SEE THAT
3114                                                            ;"CLRB" DID CLEAR
3115  015722  001411                    BEQ     1$
3116  015724  013737  001774  001124    MOV     @#UNIT, @#$GDDAT
3117  015732  052737  000100  001124    BIS     #IR,    @#$GDDAT ;GOOD DATA
3118  015740  010237  042204            MOV     R2,     @#REGADR ;FAILING REGISTER RHCS2
3119  015744  104001                    ERROR   1               ;CLEARING ODD BYTE OF RHCS2
3120                                                            ;GAVE WRONG RESULTS
3121  015746  013746  001774    1$:     MOV     @#UNIT, -(SP)
3122  015752  052716  000010            BIS     #BAI,   (SP)
3123  015756  052712  020000            BIS     #UPE,@R2        ;HAVE UPE AND MPE IN RHCS2
3124                                                            ;BESIDES UNIT SELECT
3125  015762  112612                    MOVB    (SP)+,  @P2     ;MOVE INTO EVEN BYTE OF RHCS2
3126  015764  013746  001774            MOV     @#UNIT, -(SP)
3127  015770  052716  020110            BIS     #UPE!IR!BAI,(SP)
3128  015774  011637  001124            MOV     (SP),   @#$GDDAT ;GOOD DATA
3129  016000  011237  001126            MOV     @P2,@#$BDDAT    ;TEST DATA
3130  016004  022637  001126            CMP     (SP)+,@#$BDDAT  ;COMPARE TO SEE THAT MOVB DID
3131                                                            ;MOVE EVEN BYTE ONLY
3132  016010  001403                    BEQ     TST37           ;;BRANCH IF GOOD
3133  016012  010237  042204            MOV     R2,@#REGADR     ;FAILING REGISTER RHCS2
3134  016016  104001                    ERROR   1               ;MOVING A NUMBER INTO EVEN
3135                                                            ;BYTE OF RHCS2 GAVE WRONG
3136                                                            ;RESULTS
3137
```

```
3138
3139
3140                                    ;;*****************************************************************
3141                                    ;*TEST 37       ODD BYTE TEST ON RHWC
3142
3143                                    ;**     IN THIS REGISTER NO BITS SHOULD BE PERMANENTLY SET
3144
3145                                    ;;*****************************************************************
3146  016020  000004                    TST37:  SCOPE
3147
3148  016022  012737  000037  002032            MOV     #TTNO,@#TSTNM   ;THIS SAVES TEST NUMBER
3149  016030  012706  001000                    MOV     #STACK,SP       ;RESET STACK
3150  016034  004737  042534                    JSR     PC,CLDISK       ;CLEAR DISK REGISTERS
3151  016040  013704  001632                    MOV     @#RHWC, R4      ;R4 NOW IS WORD COUNT REGISTER
3152  016044  012714  025252                    MOV     #25252, @R4     ;LOAD RHWC
3153  016050  010446                            MOV     R4,     -(SP)   ;GETTING READY TO FORM ODD BYTE
3154  016052  005216                            INC     (SP)            ;SP NOW HAS ODD BYTE FOR RHWC
3155  016054  112736  000377                    MOVB    #377,   @(SP)+  ;MOVE 377 INTO ODD BYTE OF RHWC
3156  016060  011437  001126                    MOV     @R4,@#$BDDAT    ;TEST DATA
3157  016064  022737  177652  001126            CMP     #177652,@#$BDDAT;COMPARE TO SEE IF MOVB DID OK
3158  016072  001406                            BEQ     1$              ;BRANCH IF GOOD
3159  016074  012737  177652  001124            MOV     #177652,@#$GDDAT ;GOOD DATA
3160  016102  010437  042204                    MOV     R4,     @#REGADR ;REGISTER FAILING RHWC
3161  016106  104001                            ERROR   1               ;MOVING INTO ODD BYTE OF RHWC
3162                                                                    ;GAVE WRONG RESULTS
3163  016110  112714  000123            1$:     MOVB    #123,   @R4     ;MOVE INTO EVEN BYTE OF RHWC
3164  016114  011437  001126                    MOV     @R4,@#$BDDAT    ;TEST DATA
3165  016120  022737  177523  001126            CMP     #177523,@#$BDDAT
3166  016126  001406                            BEQ     TST40           ;;BRANCH IF GOOD
3167  016130  012737  177523  001124            MOV     #177523,@#$GDDAT ;GOOD DATA
3168  016136  010437  042204                    MOV     R4,     @#REGADR ;REGISTER FAILING RHWC
3169  016142  104001                            ERROR   1
```

```
3170
3171
3172
3173                                    ;;*****************************************************************
3174                                    ;*TEST 40       TEST ODD BYTE INSTRUCTION ON RHBA
3175
3176                                    ;**     BIT 0 SHOULD ALWAYS BE 0
3177
3178                                    ;;*****************************************************************
3179  016144  000004                    TST40:  SCOPE
3180
3181  016146  012706  001000                    MOV     #STACK,SP       ;RESET STACK
3182  016152  012737  000040  002032            MOV     #TTNO,@#TSTNM   ;THIS SAVES TEST NUMBER
3183  016160  004737  042534                    JSR     PC,CLDISK
3184  016164  013704  001634                    MOV     @#RHBA, R4      ;R4 HAS ADDRESS OF RHBA
3185  016170  012714  025253                    MOV     #25253, @R4     ;LOAD RHBA
3186  016174  010446                            MOV     R4,     -(SP)   ;GETTING READY FOR ODD BYTE
3187  016176  005216                            INC     (SP)            ;SP HAS ODD BYTE ADR. OF RHBA
3188  016200  112736  000377                    MOVB    #377,   @(SP)+  ;LOAD ODD BYTE OF RHBA
3189  016204  011437  001126                    MOV     @R4,@#$BDDAT    ;TEST DATA
3190  016210  022737  177652  001126            CMP     #177652,@#$BDDAT;COMPARE MOVB RESULTS
3191  016216  001406                            BEQ     1$              ;BRANCH IF GOOD
3192  016220  012737  177652  001124            MOV     #177652,@#$GDDAT;GOOD DATA
3193  016226  010437  042204                    MOV     R4,     @#REGADR ;FAILING REGISTER RHBA
3194  016232  104001                            ERROR   1               ;MOVING INTO ODD BYTE OF
3195                                                                    ;RHBA GAVE WRONG RESULTS
3196  016234  112714  000125            1$:     MOVB    #125,   @R4
3197  016240  011437  001126                    MOV     @R4,@#$BDDAT    ;TEST DATA
3198  016244  022737  177524  001126            CMP     #177524,@#$BDDAT
3199  016252  001406                            BEQ     TST41           ;;BRANCH IF GOOD
3200  016254  012737  177524  001124            MOV     #177524,@#$GDDAT;GOOD DATA
3201  016262  010437  042204                    MOV     R4,     @#REGADR ;FAILING REGISTER RHBA
3202  016266  104001                            ERROR   1               ;MOVING INTO EVEN BYTE OF
3203                                                                    ;RHBA GAVE WRONG RESULTS
```

```
3204
3205                                    .SBTTL  DCL COMMAND TESTS
3206
3207                                    ;;*******************************************************************
3208                                    ;**     FOUR GENERAL REGISTERS WILL BE RESERVED FOR HARDWARE
3209                                    ;**     R1=RHCS1 CONTROL AND STATUS1
3210                                    ;**     R2=RHCS2 CONTROL AND STATUS2
3211                                    ;**     R3=RHDS1 DRIVE STATUS 1
3212                                    ;**     R4=RHER1 ERROR REGISTER1
3213
3214                                    ;**     WHENEVER ANY OTHER USE IS MADE OF THESE REGISTERS
3215                                    ;**     APPROPIATE SAVING MUST BE DONE
3216
3217                                    ;;*******************************************************************
3218
3219
3220
3221
3222
3223
3224                                    ;;*******************************************************************
3225                                    ;**     ERROR REGISTER #01 (RHER1) TEST
3226                                    ;**     BIT #1 (ILLEGAL REGISTER) CANNOT BE TESTED ON PDP11 THIS BIT
3227                                    ;**     IS FOR PDP10 USE ONLY
3228                                    ;;*******************************************************************
3229
3230
3231
3232                                    ;;*******************************************************************
3233                                    ;*TEST 41       TEST ILF BIT #0 IN REG. RHER1
3234
3235                                    ;**     ALL 3 ILLEGAL FUNCTION CODES SHOULD SET - ATA,ERR,ILF - AND ARE TESTED
3236                                    ;**     A GO WITHOUT CLEARING ILF ERR SHOULD SET - NXF,DLT,TRE - BITS AND THEY ARE ALSO
3237
3238                                    ;;*******************************************************************
3239  016270  000004                    TST41:  SCOPE
3240
3241  016272  012706  001000                    MOV     #STACK,SP       ;RESET STACK
3242  016276  012737  000041  002032            MOV     #TTWO,@#TSTNM   ;THIS SAVES TEST NUMBER
3243  016304  004737  042534                    JSR     PC,@#CLDISK     ;CLEAR REGISTERS
3244  016310  012777  000001  163347            MOV     #DMD,@RHMR      ;SET DIAGNOSTIC MODE
3245  016316  005037  002022                    CLR     @#TMPILL        ;GET READY TO MAKE ILLEGAL FUNCTION
3246  016322  012700  002042          1$:       MOV     #FUTABL,R0      ;LOAD FUNCTION CODE TABLE START
3247  016326  012705  000021                    MOV     #17.,R5         ;COUNTER (16 GOOD FUNCTIONS)
3248  016332  023720  002022          2$:       CMP     @#TMPILL,(R0)+  ;IS THIS A LEGAL FUNCTION CODE?
3249  016336  001004                            BNE     3$              ;NO - DECR. FUNCT. CODE CTR
3250  016340  062737  000002  002022            ADD     #2,@#TMPILL     ;YES MAKE NEXT FUNCTION CODE
3251  016346  000765                            BR      1$              ;TEST NEXT FUNCTION CODE
3252  016350  005305                  3$:       DEC     R5              ;MAKE NEXT CODE IF 1ST 16
3253                                                                    ;LEGAL FUNCTIONS NOT DONE
3254  016352  001367                            BNE     2$              ;BRANCH IF 16 NOT COMPLETE
3255  016354  032737  000100  002022            BIT     #100,@#TMPILL   ;ALL BITS UP TO BIT #5 COMPARED?
3256  016362  001077                            BNE     12$             ;YES - EXIT ------->
3257  016364  013737  002022  002104            MOV     @#TMPILL,@#ILLEGL;NO - TEST THE ILLEGAL FUNCTION
3258  016372  062737  000002  002022            ADD     #2,@#TMPILL     ;TEST NEW FUNCTION CODE NEXT TIME
3259
```

```
3260  016400  004737  042534          4$:   JSR     PC,@#CLDISK
3261  016404  012777  000001  163246        MOV     #DMD,@RHMR      ;SET DIAGNOSTIC MODE
3262  016412  013711  002104                MOV     @#ILLEGL,@R1    ;ILLEGAL FUNCTION ---> RHCS1
3263  016416  012737  016400  001110        MOV     #4$,@#$LPERR    ;ERROR RETURN POINT
3264  016424  004737  042140                JSR     PC,@#PUTREG     ;SAVE REGISTERS
3265  016430  005737  001716                TST     @#ER1           ;THERE SHOULD NOT BE ANY ERROR YET
3266  016434  001403                        BEQ     5$              ;CONTINUE IF RHER1 STILL = 0
3267  016436  010437  001122                MOV     R4,@#$BDADR     ;FAILING REGISTER ADDRESS RHER1
3268  016442  104011                        ERROR   11              ;ALTHOUGH AN ILLEGAL FUNCTION
3269                                                                ;HAS BEEN MOVED INTO RHCS1
3270                                                                ;NO ERRORS SHOULD SHOW TILL
3271                                                                ;GO IS SET RHER1 SHOULD BE
3272                                                                ;ALL ZEROS
3273
3274  016444  052711  000001          5$:   BIS     #GO,@R1         ;GO IN RHCS1
3275  016450  004737  042140                JSR     PC,@#PUTREG     ;SAVE REGISTERS
3276  016454  022737  000001  001716        CMP     #ILF,@#ER1      ;ILLEGAL FUNCTION BIT SHOULD BE SET
3277  016462  001403                        BEQ     6$              ;IT IS - CONTINUE
3278  016464  010437  001122                MOV     R4,@#$BDADR     ;FAILING REGISTER ADDRESS RHER1
3279  016470  104011                        ERROR   11              ;ILLEGAL FUNCTION DID NOT
3280                                                                ;SET ON AN ILLEGAL FUNCTION
3281                                                                ;EXECUTION, THE ILLEGAL FUNCTION
3282                                                                ;BEING EXECUTED IS IN RHCS1
3283
3284  016472  013746  001736          6$:   MOV     @#DS1,-(SP)     ;GET RHDS1
3285  016476  042716  001000                BIC     #PROG,(SP)      ;MASK PROG
3286  016502  022726  140700                CMP     #ATA!ERR!PIP!DPR!DRY,(SP)+
3287                                                                ;ATTENTION (BIT 15)
3288                                                                ;VOLUME VALID (BIT 6)
3289                                                                ;COMPOSIT ERROR (BIT 14)
3290                                                                ;DEVICE READY (BIT 7) SHOULD
3291                                                                ;BE SET ON RHDS1
3292  016506  001404                        BEQ     7$              ;THEY ARE - CONTINUE
3293  016510  013737  001662  001122        MOV     @#RHDS1,@#$RDADR ;FAILING REGISTER ADDRESS RHDS1
3294  016516  104011                        ERROR   11              ;FOLLOWING BITS SHOULD BE SET
3295                                                                ;WITH AN ILLEGAL FUNCTION
3296                                                                ;ATTENTION (BIT 15)
3297                                                                ;COMPOSIT ERROR (BIT 14)
3298                                                                ;MEDIUM ON LINE (BIT 12)
3299                                                                ;DEVICE READY (BIT 7)
3300
3301  016520  004737  044774          7$:   JSR     PC,@#MIDDLE     ;GIVE A WRITE HEADER AND
3302                                                                ;DATA COMMAND WITHOUT
3303                                                                ;CLEARING THE ERRORS
3304                                                                ;USING "MIDDLE" SO THAT
3305                                                                ;IT WILL COME BACK BEFORE
3306                                                                ;THE END TO FIND OUT ITS
3307                                                                ;STATE
3308  016524  010237  016532                MOV     R2,@#10$        ;MOVE RHCS2 ADDRESS
3309  016530  104415                        WAT                     ;WAIT FOR "NXF" BIT
3310  016532  000000                10$:    .WORD   0               ;ADDRESS OF RHCS2
3311  016534  001000                        NXF
3312  016536  004737  042140          11$:  JSR     PC,@#PUTREG     ;SAVE REGISTERS
3313
3314  016542  032737  040000  001714        BIT     #TRE,@#CS1      ;TRANSFER ERROR (BIT 14) RHCS1 - "TRE"
3315                                                                ;SHOULD SET DUE TO "NXF"
```

H7

```
3316  016550  001003                    BNE     13$             ;IT IS - CONTINUE
3317  016552  010137  001122            MOV     R1,@#SBDADR     ;FAILING REGISTER RHCS1
3318  016556  104011                    ERROR   11              ;TRANSFER ERROR (BIT 14) RHCS1 - 'TRE'
3319                                                            ;SHOULD BE SET DUE TO 'NXF'
3320                                                            ;LOCAL SCOPE RETURN POINT
3321
3322  016560  000660            13$:    BR      1$              ;GO BACK & TEST NEXT FUNCTION CODE
3323
3324  016562  000240            12$:    NOP
3325
```

```
3326
3327
3328                                    ;;*******************************************************************
3329                                    ;*TEST 42       READ IN PRESET .
3330
3331                                    ;**     ALL POSSIBLE REGISTERS WILL BE FILLED WITH ONES
3332                                    ;**     THE REGISTER CONTENTS WILL BE SAVED IN REINTO BUFFER
3333                                    ;**     THE READ IN PRESET COMMAND WILL BE GIVEN
3334                                    ;**     ALL REGISTERS WILL BE CHECKED
3335
3336                                    ;;*******************************************************************
3337  016564  000004                    TST42:  SCOPE
3338  016566  012706  001000                    MOV     #STACK,SP       ;RESET STACK
3339  016572  012737  000042  002032            MOV     #TTNO,@#TSTNM   ;THIS SAVES TEST NUMBER
3340
3341  016600  004737  042534                    JSR     PC,@#CLDISK     ;INIT AND SET GENERAL REGISTERS
3342
3343                                    ;*FILL ALL POSSIBLE BITS WITH ONES.
3344
3345  016604  012777  177777  163020            MOV     #177777,@RHWC   ;WORD COUNT REGISTER GETS 177777
3346  016612  012777  177777  163014            MOV     #177777,@RHBA   ;BUS ADDRESS REGISTER GETS 177777
3347  016620  012777  017437  163016            MOV     #17437,@RHDST   ;DESIRED SECTOR TRACK GETS 17437
3348  016626  012777  016377  163014            MOV     #16377,@RHOF    ;OFFSET REGISTER GETS 16277
3349  016634  012777  000777  163010            MOV     #777,@RHCA      ;DESIRED CYLINDER GETS 777
3350  016642  012746  001400                    MOV     #A16!A17,-(SP)  ;GET BIT 9 AND 8
3351  016646  053716  002102                    BIS     @#READIN,(SP)
3352  016652  012677  162762                    MOV     (SP)+,@RHCS1    ;FILL READ IN PRESET IN RHCS1
3353  016656  012777  000001  162774            MOV     #DMD,@RHMR      ;SET DIAGNOSTIC MODE
3354
3355                                    ;*THE REGISTERS WILL BE SAVED IN REINTO BUFFER
3356  016664  004037  043226            JSR     R0,@#SAVER      ;SAVE
3357  016670  001632                    RHWC                    ;FROM
3358  016672  003154                    REINTO                  ;TO
3359  016674  000021                    17.                     ;NUMBER SAVED
3360
3361                                    ;*GIVE READ IN PRESET COMMAND
3362  016676  052777  000001  162734    BIS     #GO,@RHCS1      ;INCLUDE GO TO READ IN PRESET
3363
3364                                    ;*NOW SAVED REGISTERS WILL BE CHANGED TO EXPECTED VALUE
3365  016704  005037  003166            CLR     @#REINTO+12     ;CLEAR SAVED RHDST
3366  016710  042737  016000  003172    BIC     #FMT22!HCI!ECI,@#REINTO+16 ;CLEAR FMT22,HCI,ECI IN
3367                                                            ;SAVED RHOF
3368  016716  052737  000100  003172    BIS     #VV,@#REINTO+16 ;SET VV IN SAVED RHOF
3369  016724  005037  003174            CLR     @#REINTO+20     ;CLEAR SAVED RHCA
3370
3371                                    ;*AFTER A READ IN PRESET COMMAND
3372                                    ;*SAVE REGISTERS AGAIN SO THAT COMPARES CAN BE DONE
3373
3374  016730  004037  043226            JSR     R0,@#SAVER      ;SAVE
3375  016734  001632                    RHWC                    ;FROM
3376  016736  002110                    WRFROM                  ;TO
3377  016740  000021                    17.                     ;NUMBER OF REGISTERS SAVED
3378
3379                                    ;*AS UPPER BYTE OF RHAS CAN BE CHANGING IN A DUAL PORT
3380                                    ;*OPERATION THE UPPER BYTE OF RHAS WILL BE SAVED AS IS
3381                                    ;*SO THAT THE COMPARES ARE ONLY VALID FOR THE LOWER BYTE
```

```
3382  016742  113737  003201  002135          MOVB    @#REINTO+25,@#WRFROM+25;SAVE UPPER RHAS
3383
3384                                          ;*COMPARE REGISTERS BEFORE READ IN PRESET COMMAND
3385                                          ;*WITH AFTER COMMAND
3386
3387  016750  004037  043430                  JSR     R0,@#COMPAR     ;COMPARE
3388  016754  003154                          REINTO                  ;GOOD BUFFER
3389  016756  002110                          WRFROM                  ;TEST BUFFER
3390  016760  000021                          17.                     ;NUMBER OF REGISTERS
3391  016762  016770                          1$                      ;RETURN FOR ERROR
3392  016764  016770                          1$                      ;SAME
3393  016766  017010                          2$                      ;RETURN FOR GOOD COMPARISON
3394
3395  016770  013705  047320          1$:     MOV     @#ERWORD,R5     ;GETTING READY TO INDEX
3396  016774  060505                          ADD     R5,R5           ;DOUBLE ERROR WORD
3397  016776  016537  001630  042204          MOV     RHWC-2(R5),@#REGADR ;FAILING REG. ADDRESS
3398  017004  104001                          ERROR   1               ;READ IN PRESET CAUSED IMPROPER
3399                                                                  ;REGISTER CHANGE
3400  017006  000207                          RTS     PC              ;RETURN FOR FURTHER COMPARISONS
3401
3402  017010                          2$:                             ;NO ERRORS
3403
```

```
3404
3405
3406                                    ;;*****************************************************************
3407                                    ;*TEST 43       NO OPERATION FUNCTION TEST
3408                                    ;**     ALL POSSIBLE REGISTERS ARE CLEARED THEN A"NOP"=0
3409                                    ;**     IS GIVEN NO CHANGE SHOULD HAPPEN
3410                                    ;**     ALL POSSIBLE REGISTERS ARE FILLED WITH ONES THEN A "NOP"
3411                                    ;**     IS GIVEN NO CHANGE SHOULD HAPPEN
3412
3413                                    ;;*****************************************************************
3414  017010  000004                    TST43:  SCOPE
3415  017012  012737  000043  002032            MOV     #TTNO,@#TSTNM   ;THIS SAVES TEST NUMBER
3416
3417                                            ;*START WITH CLR IN RHCS2 (BIT5)
3418  017020  004737  042534                    JSR     PC,@#CLDISK     ;CLEAR ALL POSSIBLE BITS
3419  017024  012777  000001  162626            MOV     #DMD,@RHMR      ;SET DIAGNOSTIC MODE
3420  017032  013711  002042                    MOV     @#NOPERA,@R1    ;PUT NOP OPERATION=0 IN RHCS1
3421  017036  012700  001632                    MOV     #RHWC,R0        ;STARTING ADDRESS OF REG
3422  017042  012703  001706                    MOV     #WC,R3          ;STARTING ADDRESS OF WHERE SAVED
3423  017046  012702  000021                    MOV     #RHEC2-RHWC+2/2,R2 ;NUMBER OF REGISTERS
3424  017052  013023                    1$:     MOV     @(R0)+,(R3)+    ;SAVE HARDWARE REG
3425  017054  005302                            DEC     R2              ;COUNT
3426  017056  001375                            BNE     1$              ;BRANCH IF NOT COMPLETE
3427  017060  013737  001662  017100            MOV     @#RHDS1,@#2$    ;GET ADDRESS OF DRIVE STATUS
3428  017066  010137  017106                    MOV     R1,@#3$         ;GET ADDRESS OF RHCS1
3429  017072  052711  000001                    BIS     #GO,@R1         ;GO TO RHCS1
3430  017076  104415                            WAT                     ;WAIT FOR DRY IN RHDS1
3431  017100  000000                    2$:     .WORD   0               ;ADDRESS OF DRIVE STATUS RHDS1
3432  017102  000200                            DRY                     ;DRY WILL BE WAITED ON
3433  017104  104415                            WAT                     ;WAIT FOR RDY IN RHCS1
3434  017106  000000                    3$:     .WORD   0               ;ADDRESS OF RHCS1 PUT HERE BY AN
3435                                            ;EARLIER MOV
3436  017110  000200                            RDY                     ;RDY WILL BE WAITED ON
3437
3438                                            ;*AFTER A NO OP  COMMAND
3439                                            ;*SAVE REGISTERS AGAIN SO THAT COMPARES CAN BE DONE
3440
3441  017112  004037  043226                    JSR     R0,@#SAVER      ;SAVE
3442  017116  001632                            RHWC                    ;FROM
3443  017120  002110                            WRFROM                  ;TO
3444  017122  000021                            17.                     ;NUMBER OF REGISTERS SAVED
3445
3446                                            ;*AS UPPER BYTE OF RHAS CAN BE CHANGING IN A DUAL PORT
3447                                            ;*OPERATION THE UPPER BYTE OF RHAS WILL BE SAVED AS IS
3448                                            ;*SO THAT THE COMPARES ARE ONLY VALID FOR THE LOWER BYTE
3449  017124  113737  001733  002135            MOVB    @#AS+1,@#WRFROM+25;SAVE UPPER RHAS
3450
3451
3452                                            ;*COMPARE REGISTERS BEFORE  NO OP COMMAND
3453                                            ;*WITH AFTER COMMAND
3454
3455  017132  004037  043430                    JSR     R0,@#COMPAR     ;COMPARE
3456  017136  001706                            WC                      ;GOOD BUFFER
3457  017140  002110                            WRFROM                  ;TEST BUFFER
3458  017142  000021                            17.                     ;NUMBER OF REGISTERS
3459  017144  017152                            4$                      ;RETURN FOR ERROR
```

```
3460  017146  017152                          4$                      ;SAME
3461  017150  017172                          5$                      ;RETURN FOR GOOD COMPARISON
3462
3463  017152  013705  047320          4$:     MOV     @#ERWORD,R5     ;GETTING READY TO INDEX
3464  017156  060505                          ADD     R5,R5           ;DOUBLE ERROR WORD
3465  017160  016537  001630  042204          MOV     RHWC-2(R5),@#REGADR ;FAILING REG. ADDRESS
3466  017166  104001                          ERROR   1               ;NO OP COMMAND CAUSED IMPROPER
3467                                                                  ;REGISTER CHANGE
3468  017170  000207                          RTS     PC              ;RETURN FOR FURTHER COMPARISONS
3469
3470  017172                          5$:                             ;NO ERRORS
3471
3472
3473
3474  017172  012737  017200  001110          MOV     #14$,@#$LPERR   ;SET SCOPE LOOP TO 14$
3475  017200  004737  042534          14$:    JSR     PC,@#CLDISK     ;INIT LAST ALL ZERO TEST
3476  017204  012777  000001  162446          MOV     #DMD,@RHMR      ;SET DIAGNOSTIC MODE
3477
3478
3479                                          ;*NOW START WITH ALL ONES IN ALL POSSIBLE REGISTERS
3480
3481  017212  012700  001632                  MOV     #RHWC,R0        ;ADDRESS OF FIRST REGISTER
3482  017216  012705  000021                  MOV     #RHEC2-RHWC+2/2,R5 ;NO. OF REGISTERS
3483  017222  012730  177676          6$:     MOV     #177676,@(R0)+  ;FILL WITH ALL ONES
3484  017226  013777  001774  162402          MOV     @#UNIT,@RHCS2   ;REINSTATE UNIT NUMBER UNDER TEST
3485                                                                  ;KEEP INTERRUPT DISABLED
3486  017234  005305                          DEC     R5              ;COUNT
3487  017236  001371                          BNE     6$              ;BRANCH IF INCOMPLETE
3488  017240  013711  002042                  MOV     @#NOPERA,@R1    ;PUT NOP OPERATION =0 IN RHCS1
3489  017244  012700  001632                  MOV     #RHWC,R0        ;STARTING ADDRESS OF REG
3490  017250  012703  001706                  MOV     #WC,R3          ;STARTING ADDRESS OF WHERE SAVED
3491  017254  012702  000021                  MOV     #RHEC2-RHWC+2/2,R2 ;NUMBER OF REGISTERS
3492  017260  013023                  7$:     MOV     @(R0)+,(R3)+    ;SAVE HARDWARE REG
3493  017262  005302                          DEC     R2              ;COUNT
3494  017264  001375                          BNE     7$              ;BRANCH IF NOT COMPLETE
3495  017266  013737  001662  017306          MOV     @#RHDS1,@#10$   ;GET ADDRESS OF DRIVE STATUS
3496  017274  010137  017314                  MOV     R1,@#11$        ;GET ADDRESS OF RHCS1
3497  017300  052711  000001                  BIS     #GO,@R1         ;GO TO RHCS1
3498  017304  104415                          WAT                     ;WAIT FOR DRY IN RHDS1
3499  017306  000000                  10$:    .WORD   0               ;ADDRESS OF DRIVE STATUS RHDS1
3500  017310  000200                          DRY                     ;DRY WILL BE WAITED ON
3501  017312  104415                          WAT                     ;WAIT FOR RDY IN RHCS1
3502  017314  000000                  11$:    .WORD   0               ;ADDRESS OF RHCS1 PUT HERE BY AN
3503                                                                  ;EARLIER MOV.
3504  017316  000200                          RDY                     ;RDY WILL BE WAITED ON
3505
3506                                          ;*AFTER A NO OP COMMAND
3507                                          ;*SAVE REGISTERS AGAIN SO THAT COMPARES CAN BE DONE
3508
3509  017320  004037  043226                  JSR     R0,@#SAVER      ;SAVE
3510  017324  001632                          RHWC                    ;FROM
3511  017326  002110                          WRFROM                  ;TO
3512  017330  000021                          17.                     ;NUMBER OF REGISTERS SAVED
3513
3514                                          ;*AS UPPER BYTE OF RHAS CAN BE CHANGING IN A DUAL PORT
3515                                          ;*OPERATION THE UPPER BYTE OF RHAS WILL BE SAVED AS IS
```

M7

```
3516                                        ;*SO THAT THE COMPARES ARE ONLY VALID FOR THE LOWER BYTE
3517  017332  113737  001733  002135        MOVB    @#AS+1,@#WRFROM+25;SAVE UPPER RHAS
3518
3519
3520                                        ;*COMPARE REGISTERS BEFORE NO OP COMMAND
3521                                        ;*WITH AFTER COMMAND
3522
3523  017340  004037  043430                JSR     R0,@#COMPAR     ;COMPARE
3524  017344  001706                        WC                      ;GOOD BUFFER
3525  017346  002110                        WRFROM                  ;TEST BUFFER
3526  017350  000021                        17.                     ;NUMBER OF REGISTERS
3527  017352  017360                        12$                     ;RETURN FOR ERROR
3528  017354  017360                        12$                     ;SAME
3529  017356  017400                        13$                     ;RETURN FOR GOOD COMPARISON
3530
3531  017360  013705  047320        12$:    MOV     @#ERWORD,R5     ;GETTING READY TO INDEX
3532  017364  060505                        ADD     R5,R5           ;DOUBLE ERROR WORD
3533  017366  016537  001630  042204        MOV     RHWC-2(R5),@#REGADR ;FAILING REG. ADDRESS
3534  017374  104001                        ERROR   1               ;NO OP COMMAND CAUSED IMPROPER
3535                                        ;REGISTER CHANGE
3536  017376  000207                        RTS     PC              ;RETURN FOR FURTHER COMPARISONS
3537
3538  017400                        13$:                            ;NO ERRORS
```

```
3539
3540
3541                                    ;;*******************************************************************
3542                                    ;*TEST 44       DRIVE CLEAR
3543
3544                                    ;**     ALL WRITE BITS OF ALL REGISTERS EXCEPT RHDB ARE FILLED WITH
3545                                    ;**     ONES EXCEPT FOR BIT #0 AND BIT #6 WHICH ARE "GO" AND
3546                                    ;**     "ENABLE INTERRUPT" BITS
3547                                    ;**     THEN A DRIVE CLEAR IS PERFORMED
3548                                    ;**     THEN ALL REGISTERS EXCEPT RHDB ARE CHECKED
3549
3550                                    ;;*******************************************************************
3551  017400  000004                    TST44:  SCOPE
3552  017402  012706  001000                    MOV     #STACK,SP       ;RESET STACK
3553  017406  012737  000044  002032            MOV     #TTNO,@#TSTNM   ;THIS SAVES TEST NUMBER
3554  017414  004737  042534                    JSR     PC,@#CLDISK     ;SET REGISTERS AND CLEAR
3555
3556                                            ;*FILL ALL POSSIBLE BITS WITH ONES
3557
3558  017420  012777  177777  162202           MOV     #177777,@RHDB   ;BUS ADDRESS REGISTER GETS 177777
3559  017426  012777  177777  162176           MOV     #177777,@RHWC   ;WORD COUNT REGISTER GETS 177777
3560  017434  012777  177777  162172           MOV     #177777,@RHBA   ;BUS ADDRESS REGISTER GETS 177777
3561  017442  052777  157010  162166           BIS     #157010,@RHCS2  ;CONTROL AND STATUS 2 GETS 157010
3562  017450  012777  001476  162162           MOV     #1476,@RHCS1    ;CONTROL AND STATUS REGISTER/GETS 1476
3563  017456  012777  177777  162156           MOV     #177777,@RHER1  ;ERROR REGISTER1 GETS 177777
3564  017464  012777  017437  162152           MOV     #17437,@RHDST   ;DESIRED SECTOR TRACK
3565  017472  012777  177777  162146           MOV     #177777,@RHER2  ;ERROR REGISTER 2
3566  017500  012777  016277  162142           MOV     #16277,@RHOF    ;OFFSET REGISTER
3567  017506  012777  177777  162136           MOV     #177777,@RHCA   ;DESIRED CYLINDER
3568  017514  012777  177777  162132           MOV     #177777,@RHER3  ;ERROR REGISTER 3
3569  017522  012777  000001  162130           MOV     #DMD,@RHMR      ;MAINTENANCE REGISTER
3570  017530  012777  177777  162122           MOV     #177777,@RHMR   ;MAINTENANCE REGISTER
3571
3572
3573                                            ;*THIS SETS BITS FOR ALL PRESENT DRIVES
3574
3575  017536  013700  002020                    MOV     @#TOTALAT,R0    ;GET DRIVE PRESENT
3576  017542  005012                            CLR     @R2             ;CLEAR RHCS2 AND CARRY BIT
3577  017544  012705  000010                    MOV     #8.,R5          ;COUNTER
3578  017550  006000                  30$:      ROR     R0              ;GET BIT INTO CARRY
3579  017552  103002                            BCC     31$             ;BRANCH IF NO UNIT ON THIS BIT
3580  017554  012714  177777                    MOV     #-1,@R4         ;MOVE INTO ERROR REGISTER TO SET ATA
3581  017560  005212                  31$:      INC     @R2             ;INCREMENT RHCS2 - UNIT NO.
3582  017562  005305                            DEC     R5              ;COUNT
3583  017564  001401                            BEQ     27$             ;BRANCH IF 8 DONE
3584  017566  000770                            BR      30$             ;CONTINUE THIS ROUTINE
3585  017570  013746  001774          27$:      MOV     @#UNIT,-(SP)
3586  017574  052716  157010                    BIS     #157010,(SP)    ;REINSTATE SET BITS
3587  017600  012612                            MOV     (SP)+,@R2       ;
3588
3589
3590  017602  012777  000001  162050           MOV     #DMD,@RHMR      ;SET DMD
3591  017610  013711  002050                    MOV     @#DCLEAR,@R1    ;DRIVE CLEAR = 10 INTO RHCS1
3592  017614  052711  000001                    BIS     #GO,@R1         ;GO
3593  017620  012700  001630                    MOV     #RHDB,  R0      ;R0 CONTAINS ADDR. OF ADDR. OF REG.
3594
```

```
3595
3596                                        ;*DATA BUFFER REGISTER
3597
3598
3599  017624  012737  177777  001124  28$:  MOV   #177777,@#$GDDAT      ;GOOD DATA FOR ERROR TYPEOUT
3600  017632  013037  001126                MOV   @(R0)+,@#$BDDAT ;TEST DATA
3601  017636  022737  177777  001126        CMP   #177777,@#$BDDAT      ;COMPARE DATA
3602  017644  001402                        BEQ   3$              ;BRANCH IF GOOD
3603  017646  004737  020504                JSR   PC,@#ERCLFC     ;JUMP TO ERROR FOR CLR (BIT 5)
3604                                                              ;IN RHCS2
3605
3606
3607                                        ;*WORD COUNT REGISTER
3608
3609
3610  017652  012737  177777  001124  3$:   MOV   #177777,@#$GDDAT      ;GOOD DATA FOR ERROR TYPEOUT
3611  017660  013037  001126                MOV   @(R0)+,@#$BDDAT ;TEST DATA
3612  017664  022737  177777  001126        CMP   #177777,@#$BDDAT      ;COMPARE DATA
3613  017672  001402                        BEQ   4$              ;BRANCH IF GOOD
3614  017674  004737  020504                JSR   PC,@#ERCLFC     ;JUMP TO ERROR FOR CLR (BIT 5)
3615                                                              ;IN RHCS2
3616
3617
3618                                        ;*BUS ADDRESS REGISTER
3619
3620
3621  017700  012737  177776  001124  4$:   MOV   #177776,@#$GDDAT      ;GOOD DATA FOR ERROR TYPEOUT
3622  017706  013037  001126                MOV   @(R0)+,@#$BDDAT ;TEST DATA
3623  017712  022737  177776  001126        CMP   #177776,@#$BDDAT      ;COMPARE DATA
3624  017720  001402                        BEQ   5$              ;BRANCH IF GOOD
3625  017722  004737  020504                JSR   PC,@#ERCLFC     ;JUMP TO ERROR FOR CLR (BIT 5)
3626                                                              ;IN RHCS2
3627
3628
3629                                        ;*CONTROL AND STATUS 2 REGISTER
3630
3631
3632
3633  017726  012746  000110          5$:   MOV   #110,-(SP)      ;INCLUDE IR
3634  017732  053716  001774                BIS   @#UNIT,(SP)     ;SET UNIT NO.
3635  017736  012637  001124                MOV   (SP)+,@#$GDDAT  ;GOOD DATA FOR TYPE OUT
3636  017742  013037  001126                MOV   @(R0)+,@#$BDDAT ;TEST DATA
3637  017746  023737  001124  001126        CMP   @#$GDDAT,@#$BDDAT;COMPARE DATA
3638  017754  001402                        BEQ   6$              ;BRANCH IF GOOD
3639  017756  004737  020504                JSR   PC,@#ERCLFC     ;JUMP TO ERROR FOR CLR (BIT 5)
3640                                                              ;IN RHCS2
3641
3642
3643
3644                                        ;*CONTROL AND STATUS 1 REGISTER
3645
3646  017762  005737  002000          6$:   TST   @#NUNIT         ;ARE THERE MORE THAN ONE UNIT
3647  017766  001404                        BEQ   32$             ;BRANCH IF ONLY ONE UNIT
3648  017770  012737  104210  001124        MOV   #104210,@#$GDDAT;GOOD DATA
3649  017776  000403                        BR    33$
3650  020000  012737  004210  001124  32$:  MOV   #4210,@#$GDDAT  ;GOOD DATA
```

```
3651  020006  013037  001126          33$:    MOV     @(R0)+,@#$BDDAT ;TEST DATA
3652
3653  020012  023737  001124  001126          CMP     @#$GDDAT,@#$BDDAT;COMPARE DATA
3654  020020  001402                          BEQ     7$              ;BRANCH IF GOOD
3655  020022  004737  020504                  JSR     PC,@#ERCLFC     ;JUMP TO ERROR FOR CLR BIT 5
3656                                                                  ;IN RHCS2
3657
3658                                          ;*ERROR 1 REGISTER
3659
3660
3661  020026  012737  000000  001124  7$:     MOV     #0,@#$GDDAT     ;GOOD DATA FOR ERROR TYPEOUT
3662  020034  013037  001126                  MOV     @(R0)+,@#$BDDAT ;TEST DATA
3663  020040  022737  000000  001126          CMP     #0,@#$BDDAT     ;COMPARE DATA
3664  020046  001402                          BEQ     10$             ;BRANCH IF GOOD
3665  020050  004737  020504                  JSR     PC,@#ERCLFC     ;JUMP TO ERROR FOR CLR (BIT 5)
3666                                                                  ;IN RHCS2
3667
3668                                          ;*DESIRED SECTOR/TRACK REGISTER
3669
3670
3671  020054  012737  017437  001124  10$:     MOV     #17437,@#$GDDAT         ;GOOD DATA FOR ERROR TYPEOUT
3672  020062  013037  001126                  MOV     @(R0)+,@#$BDDAT ;TEST DATA
3673  020066  022737  017437  001126          CMP     #17437,@#$BDDAT ;COMPARE DATA
3674  020074  001402                          BEQ     11$             ;BRANCH IF GOOD
3675  020076  004737  020504                  JSR     PC,@#ERCLFC     ;JUMP TO ERROR FOR CLR (BIT 5)
3676                                                                  ;IN RHCS2
3677
3678                                          ;*ERROR 2 REGISTER
3679
3680
3681  020102  012737  000000  001124  11$:     MOV     #0,@#$GDDAT     ;GOOD DATA FOR ERROR TYPEOUT
3682  020110  013037  001126                  MOV     @(R0)+,@#$BDDAT ;TEST DATA
3683  020114  022737  000000  001126          CMP     #0,@#$BDDAT     ;COMPARE DATA
3684  020122  001402                          BEQ     12$             ;BRANCH IF GOOD
3685  020124  004737  020504                  JSR     PC,@#ERCLFC     ;JUMP TO ERROR FOR CLR (BIT 5)
3686                                                                  ;IN RHCS2
3687
3688                                          ;*OFFSET REGISTER
3689
3690
3691  020130  012737  116000  001124  12$:     MOV     #116000,@#$GDDAT        ;GOOD DATA FOR ERROR TYPEOUT
3692  020136  013037  001126                  MOV     @(R0)+,@#$BDDAT ;TEST DATA
3693  020142  022737  116000  001126          CMP     #116000,@#$BDDAT        ;COMPARE DATA
3694  020150  001402                          BEQ     13$             ;BRANCH IF GOOD
3695  020152  004737  020504                  JSR     PC,@#ERCLFC     ;JUMP TO ERROR FOR CLR (BIT 5)
3696                                                                  ;IN RHCS2
3697
3698                                          ;*DESIRED CYLINDER ADDRESS REGISTER
3699
3700
3701  020156  012737  001777  001124  13$:     MOV     #1777,@#$GDDAT  ;GOOD DATA FOR ERROR TYPEOUT
3702  020164  013037  001126                  MOV     @(R0)+,@#$BDDAT ;TEST DATA
3703  020170  022737  001777  001126          CMP     #1777,@#$BDDAT  ;COMPARE DATA
3704  020176  001402                          BEQ     14$             ;BRANCH IF GOOD
3705  020200  004737  020504                  JSR     PC,@#ERCLFC     ;JUMP TO ERROR FOR CLR (BIT 5)
3706                                                                  ;IN RHCS2
```

```
3707
3708                                        ;*ERROR 3 REGISTER
3709
3710
3711  020204  012737  000000  001124  14$:  MOV     #0,@#$GDDAT     ;GOOD DATA FOR ERROR TYPEOUT
3712  020212  013037  001126                MOV     @(R0)+,@#$BDDAT ;TEST DATA
3713  020216  022737  000000  001126        CMP     #0,@#$BDDAT     ;COMPARE DATA
3714  020224  001402                        BEQ     15$             ;BRANCH IF GOOD
3715  020226  004737  020504                JSR     PC,@#ERCLFC     ;JUMP TO ERROR FOR CLR (BIT 5)
3716                                                                ;IN RHCS2
3717
3718                                        ;*ATTENTION SUMMARY REGISTER
3719
3720  020232  013737  002020  001124  15$:  MOV     @#TOTALAT,@#$GDDAT;SET ALL BITS OF DRIVE PRESENT IN RHAS
3721  020240  043737  002016  001124        BIC     @#ATTENT,@#$GDDAT       ;CLEAR ONLY WORKING DRIVE BIT
3722  020246  013037  001126                MOV     @(R0)+,@#$BDDAT         ;GET RHAS
3723  020252  123737  001124  001126        CMPB    @#$GDDAT,@#$BDDAT       ;COMPARE DATA
3724  020260  001402                        BEQ     16$                     ;BRANCH IF GOOD
3725  020262  004737  020504                JSR     PC,@#ERCLFC             ;JUMP TO ERROR FOR CLR (BIT 5) IN RHCS2
3726
3727                                        ;*MAINTAINABILITY REGISTER
3728
3729
3730  020266  012737  000400  001124  16$:  MOV     #400,@#$GDDAT   ;GOOD DATA FOR ERROR TYPEOUT
3731  020274  013037  001126                MOV     @(R0)+,@#$BDDAT ;TEST DATA
3732  020300  022737  000400  001126        CMP     #400,@#$BDDAT   ;COMPARE DATA
3733  020306  001402                        BEQ     17$             ;BRANCH IF GOOD
3734  020310  004737  020504                JSR     PC,@#ERCLFC     ;JUMP TO ERROR FOR CLR (BIT 5)
3735                                                                ;IN RHCS2
3736
3737                                        ;*DRIVE STATUS REGISTER
3738
3739  020314  012737  000700  001124  17$:  MOV     #700,@#$GDDAT   ;GOOD DATA FOR PRINTOUT
3740  020322  013046                        MOV     @(R0)+,-(SP)    ;GET RHDS1
3741  020324  011637  001126                MOV     (SP),@#$BDDAT   ;TEST DATA
3742  020330  042716  001000                BIC     #PROG,(SP)      ;CLEAR PROG BIT
3743  020334  022726  000700                CMP     #700,(SP)+      ;COMPARE DATA
3744  020340  001402                        BEQ     20$             ;BRANCH IF GOOD
3745  020342  004737  020504                JSR     PC,@#ERCLFC     ;JUMP TO ERROR FOR DRIVE CLEAR
3746
3747                                        ;*DRIVE TYPE
3748
3749
3750  020346  013737  002010  001124  20$:  MOV     @#SAVDT,@#$GDDAT        ;GOOD DATA FOR ERROR TYPEOUT
3751  020354  013037  001126                MOV     @(R0)+,@#$BDDAT ;TEST DATA
3752  020360  023737  002010  001126        CMP     @#SAVDT,@#$BDDAT        ;COMPARE DATA
3753  020366  001402                        BEQ     21$             ;BRANCH IF GOOD
3754  020370  004737  020504                JSR     PC,@#ERCLFC     ;JUMP TO ERROR FOR CLR (BIT 5)
3755                                                                ;IN RHCS2
3756
3757                                        ;*SERIAL NUMBER REGISTER
3758
3759
3760
3761  020374  013737  002012  001124  21$:  MOV     @#SAVSN,@#$GDDAT        ;GOOD DATA FOR ERROR TYPEOUT
3762  020402  013037  001126                MOV     @(R0)+,@#$BDDAT ;TEST DATA
```

```
3763  020406  023737  002012  001126          CMP     @#SAVSN,@#$BDDAT        ;COMPARE DATA
3764  020414  001402                          BEQ     22$             ;BRANCH IF GOOD
3765  020416  004737  020504                  JSR     PC,@#ERCLFC     ;JUMP TO ERROR FOR CLR (BIT 5)
3766                                                                  ;IN RHCS2
3767
3768                                          ;*ECC1 POSITION
3769
3770
3771  020422  012737  000000  001124  22$:    MOV     #0,@#$GDDAT     ;GOOD DATA FOR ERROR TYPEOUT
3772  020430  013037  001126                  MOV     @(R0)+,@#$BDDAT ;TEST DATA
3773  020434  022737  000000  001126          CMP     #0,@#$BDDAT     ;COMPARE DATA
3774  020442  001402                          BEQ     23$             ;BRANCH IF GOOD
3775  020444  004737  020504                  JSR     PC,@#ERCLFC     ;JUMP TO ERROR FOR CLR (BIT 5)
3776                                                                  ;IN RHCS2
3777
3778                                          ;*ECC2 PATTERN
3779
3780
3781  020450  012737  000000  001124  23$:    MOV     #0,@#$GDDAT     ;GOOD DATA FOR ERROR TYPEOUT
3782  020456  013037  001126                  MOV     @(R0)+,@#$BDDAT ;TEST DATA
3783  020462  022737  000000  001126          CMP     #0,@#$BDDAT     ;COMPARE DATA
3784  020470  001402                          BEQ     24$             ;BRANCH IF GOOD
3785  020472  004737  020504                  JSR     PC,@#ERCLFC     ;JUMP TO ERROR FOR CLR (BIT 5)
3786                                                                  ;IN RHCS2
3787
3788
3789                                          ;*LOOK-AHEAD REGISTER
3790
3791  020476  005720                  24$:    TST     (R0)+           ;AS THE LOOK-AHEAD REG. CANNOT BE PREDICTED
3792                                                                  ;IT IS NOT CHECKED AFTER AN INIT
3793
3794                                          ;*CURRENT CYLINDER ADDRESS REGISTER
3795
3796  020500  005720                  25$:    TST     (R0)+           ;AS THE CURRENT CYL REG. CANNOT BE PREDICTED
3797                                                                  ;AFTER AN INIT IT IS NOT CHECKED
3798
3799  020502                          26$:
3800  020502  000413                          BR      TST45           ;BRANCH OVER JSR
3801
3802  020504  014037  042204          ERCLFC: MOV     -(R0),  @#REGADR ;FAILING REGISTER ADDRESS
3803  020510  104001                          ERROR   1               ;CLR FUNCTION = 10 IN RHCS1 DID
3804                                                                  ;NOT CLEAR APPROPIATE BITS
3805                                                                  ;OR CLEARED EXTRA BITS
3806  020512  005720                          TST     (R0)+           ;UNDO -(R0) FOR ERROR
3807  020514  000207                          RTS     PC              ;RETURN TO ABOVE PROGRAM
3808                                                                  ;OR CLEARED EXTRA BITS
3809  020516  005720                          TST     (R0)+           ;UNDO -(R0) FOR BAD DATA
3810  020520  004737  042570                  JSR     PC,@#CHECKT     ;CHECK THAT DVA,RDY,DPR,DRY = 1
3811  020524  104401  062145                  TYPE    ,CPHALT         ;AND THAT NO OTHERS = 1.  CANNOT CON-
3812                                                                  ;TINUE TESTING IF BOTH AREN'T TRUE
3813  020530  000000                          HALT                    ;STOP THE TEST
3814
```

```



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

                                        ;;*******************************************************************
 020532  000004                         TST45:  SCOPE
 020534  012706  001000                         MOV     #STACK,SP           ;RESET STACK
 020540  012737  000045  002032                 MOV     #TTNO,@#TSTNM       ;THIS SAVES TEST NUMBER
 020546  004737  042534                         JSR     PC,@#CLDISK         ;INIT AND SET UP GENERAL REG. CORRES.
                                                                            ;AND UNIT NUMBER
 020552  012777  000001  161100                 MOV     #DMD,@RHMR          ;SET DIAGNOSTIC MODE BIT
                                                                            ;THIS ENABLES COMMANDS WITHOUT MOL
                                                                            ;AND HOLDS RHLA FROM MOVING
 020560  005077  161060                         CLR     @RHDST              ;MAKE DESIRED SECTOR TRACK LEGAL
 020564  013777  002072  161046                 MOV     @#SEECOM,@RHCS1     ;LOAD SEEK COMMAND INTO CONTROLLER
 020572  017746  161100                         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

 020576  005737  002036                         TST     @#RP06   ;MOVE DRIVE TYPE FLAG TO ITSELF TO TEST
 020602  001416                                 BEQ     11$                 ;TREAT THE DRIVE AS AN RP04

                                                                            ;TREAT THE DRIVE AS AN RP06
 020604  022726  001456                         CMP     #814.,(SP)+         ;IS CURRENT CYLINDER SAME AS 814. ?
 020610  001404                                 BEQ     9$                  ;BRANCH IF YES TO MAKE RHCA = 813.
 020612  012737  001456  001210                 MOV     #814.,@#$TMP5       ;GET READY TO MAKE RHCA = 814.
 020620  000403                                 BR      10$                 ;FILL RHCA
 020622  012737  001455  001210         9$:     MOV     #813.,@#$TMP5       ;GET READY TO MAKE RHCA = 813.
 020630  013777  001210  161014         10$:    MOV     @#$TMP5,@RHCA       ;MAKE DESIRED CYLINDER 814., OR 813.
 020636  000415                                 BR      14$                 ;SAVE REGISTERS

                                                                            ;TREAT THE DRIVE AS AN RP04
 020640  022726  000632                 11$:    CMP     #410.,(SP)+         ;IS CURRENT CYLINDER SAME AS 410. ?
 020644  001404                                 BEQ     12$                 ;BRANCH IF YES TO MAKE RHCA = 409.
 020646  012737  000632  001210                 MOV     #410.,@#$TMP5       ;GET READY TO MAKE RHCA = 410.
 020654  000403                                 BR      13$                 ;FILL RHCA
 020656  012737  000631  001210         12$:    MOV     #409.,@#$TMP5       ;GET READY TO MAKE RHCA = 409.
 020664  013777  001210  160760         13$:    MOV     @#$TMP5,@RHCA       ;MAKE DESIRED CYLINDER 410., OR 409.

                                                ;*SAVE REGISTERS FOR COMPARISON AFTER GO

 020672  004037  043226                 14$:    JSR     R0,@#SAVER          ;SAVE
 020676  001632                                 RHWC                        ;FROM
 020700  003154                                 REINTO                      ;TO
 020702  000023                                 19.                         ;NUMBER OF REGISTERS SAVED

                                                ;*GIVE GO TO COMMAND
```

69

```
3871  020704  052777  000001  160726          BIS     #GO,@RHCS1      ;GO TO COMMAND
3872
3873                                          ;*CHANGE SAVED REGISTERS TO EXPECTED VALUES
3874
3875  020712  052737  000001  003162          BIS     #GO,@#REINTO+6  ;SAVED RHCS1
3876  020720  052737  020000  003204          BIS     #PIP,@#REINTO+30 ;SAVED RHDS1
3877  020726  042737  000200  003204          BIC     #DRY,@#REINTO+30 ;SAVED RHDS1
3878
3879
3880                                          ;*AFTER GO HAS BEEN GIVEN FOR SEEK COMMAND
3881                                          ;*SAVE REGISTERS AGAIN SO THAT COMPARISONS CAN
3882                                          ;*BE DONE
3883
3884  020734  004037  043226                  JSR     R0,@#SAVER      ;SAVE
3885  020740  001632                          RHWC                    ;FROM
3886  020742  002110                          WRFROM                  ;TO
3887  020744  000023                          19.                     ;NUMBER OF REGISTERS SAVED
3888
3889                                          ;*AS UPPER BYTE OF RHAS CAN BE CHANGING IN A DUAL PORT
3890                                          ;*OPERATION THE UPPER BYTE OF RHAS WILL BE SAVED AS IS
3891                                          ;*SO THAT THE COMPARES ARE ONLY VALID FOR THE LOWER BYTE
3892
3893  020746  113737  003201  002135          MOVB    @#REINTO+25,@#WRFROM+25;SAVE UPPER RHAS
3894
3895                                          ;*COMPARE REGISTERS BEFORE SEEK COMMAND
3896                                          ;*WITH CONTENTS AFTER GO IS ISSUED
3897
3898  020754  004037  043430                  JSR     R0,@#COMPAR     ;COMPARE
3899  020760  003154                          REINTO                  ;GOOD BUFFER
3900  020762  002110                          WRFROM                  ;TEST BUFFER
3901  020764  000023                          19.                     ;NUMBER
3902  020766  020774                          1$                      ;RETURN FOR ERROR
3903  020770  020774                          1$                      ;SAME
3904  020772  021014                          2$                      ;RETURN FOR GOOD COMPARISON
3905
3906  020774  013705  047320          1$:     MOV     @#ERWORD,R5     ;GETTING READY TO INDEX
3907  021000  060505                          ADD     R5,R5           ;DOUBLE ERROR WORD
3908  021002  016537  001630  042204          MOV     RHWC-2(R5),@#REGADR ;FAILING REGISTER ADDRESS
3909
3910  021010  104001                          ERROR   1               ;IMPROPER REGISTER CHANGE
3911                                                                  ;AFTER SEEK COMMAND
3912                                                                  ;WITH GO IS GIVEN
3913  021012  000207                          RTS     PC              ;RETURN TO COMPARISON
3914
3915
3916                                          ;*NOW GIVE INIT AND GET GO AND PIP DOWN
3917
3918  021014  052712  000040          2$:     BIS     #CLR,@R2        ;RH INITILIZE
3919  021020  013712  001774                  MOV     @#UNIT,@R2      ;REINSTATE UNIT NUMBER
3920  021024  012777  000001  160626          MOV     #DMD,@RHMR      ;SET DIAGNOSTIC MODE BIT
3921                                                                  ;THIS ENABLES COMMANDS WITHOUT MOL
3922                                                                  ;AND HOLDS RHLA FROM MOVING
3923
3924                                          ;*CHANGE REGISTERS TO EXPECTED VALUE
3925
3926  021032  042737  000001  003162          BIC     #GO,@#REINTO+6  ;SAVED RHCS1
```

```
3927  021040  042737  020000  003204          BIC     @#PIP,@#REINTO+30 ;SAVED RHDS1
3928  021046  052737  000200  003204          BIS     @#DRY,@#REINTO+30 ;SAVED RHDS1
3929  021054  017737  160614  003216          MOV     @RHLA,@#REINTO+42;SAVED RHLA
3930  021062  013737  001210  003220          MOV     @#$TMP5,@#REINTO+44 ;SAVED RHCC
3931
3932
3933                                          ;*AFTER INITILIZE SAVE REGISTERS SO THAT
3934                                          ;*COMPARES CAN BE DONE
3935
3936  021070  004037  043226                  JSR     R0,@#SAVER      ;SAVE
3937  021074  001632                          RHWC                    ;FROM
3938  021076  002110                          WRFROM                  ;TO
3939  021100  000023                          19.                     ;NUMBER OF REGISTERS SAVED
3940
3941
3942                                          ;*AS UPPER BYTE OF RHAS CAN BE CHANGING IN A DUAL PORT
3943                                          ;*OPERATION THE UPPER BYTE OF RHAS WILL BE SAVED AS IS
3944                                          ;*SO THAT THE COMPARES ARE ONLY VALID FOR THE LOWER BYTE
3945
3946  021102  113737  003201  002135          MOVB    @#REINTO+25,@#WRFROM+25;SAVE UPPER RHAS
3947
3948
3949                                          ;*COMPARE REGISTERS AFTER INITIALIZE
3950
3951  021110  004037  043430                  JSR     R0,@#COMPAR     ;COMPARE
3952  021114  003154                          REINTO                  ;GOOD BUFFER
3953  021116  002110                          WRFROM                  ;TEST BUFFER
3954  021120  000023                          19.                     ;NUMBER OF REGISTERS TO BE
3955                                                                  ;COMPARED
3956  021122  021130                          3$                      ;RETURN POINT FOR ERROR
3957  021124  021130                          3$                      ;SAME
3958  021126  021150                          4$                      ;RETURN POINT FOR GOOD COMPARISON
3959
3960  021130  013705  047320          3$:     MOV     @#ERWORD,R5     ;GETTING READY TO INDEX
3961  021134  060505                          ADD     R5,R5           ;DOUBLE ERROR WORD
3962  021136  016537  001630  042204          MOV     RHWC-2(R5),@#REGADR ;FAILING REGISTER ADDRESS
3963  021144  104001                          ERROR   1               ;IMPROPER REGISTER
3964                                                                  ;CONTENTS AFTER GIVING AN
3965                                                                  ;INITILIZE FOLLOWING A
3966                                                                  ;SEEK COMMAND
3967  021146  000207                          RTS     PC              ;RETURN TO COMPARISON
3968
3969  021150                          4$:                             ;GOOD COMPARISON
3970
3971  021150  004737  042534                  JSR     PC,@#CLDISK     ;INIT AND SET UP GENERAL REG.
3972                                                                  ;AND UNIT NUMBER
3973  021154  012777  000001  160476          MOV     #DMD,@RHMR      ;SET DIAGNOSTIC MODE BIT
3974                                                                  ;THIS ENABLES COMMANDS WITHOUT MOL
3975                                                                  ;AND HOLDS RHLA FROM MOVING
3976
3977  021162  013777  002072  160450          MOV     @#SEECOM,@RHCS1 ;LOAD SEEK COMMAND INTO RHC
3978  021170  005077  160456                  CLR     @RHCA           ;DESIRED CYLINDER ADDRESS
3979
3980                                          ;*SAVE REGISTERS FOR COMPARISON AFTER GO
3981  021174  004037  043226                  JSR     R0,@#SAVER      ;SAVE
3982  021200  001632                          RHWC                    ;FROM
```

```
3983  021202  003154                          REINTO                  ;TO
3984  021204  000023                          19.                     ;NUMBER OF REGISTERS SAVED
3985
3986                                          ;*GIVE GO TO SEEK COMMAND
3987  021206  052777  000001  160424          BIS     #GO,@RHCS1      ;GO TO SEEK COMMAND
3988
3989                                          ;*CHANGE SAVED REGISTERS TO EXPECTED VALUES
3990
3991  021214  052737  000001  003162          BIS     #GO,@#REINTO+6  ;SAVED RHCS1
3992  021222  052737  020000  003204          BIS     #PIP,@#REINTO+30 ;SAVED RHDS1
3993  021230  042737  000200  003204          BIC     #DRY,@#REINTO+30 ;SAVED RHDS1
3994
3995
3996                                          ;*AFTER GO HAS BEEN GIVEN TO SEEK COMMAND
3997                                          ;*SAVE REGISTERS AGAIN SO THAT COMPARISONS CAN
3998                                          ;*BE DONE
3999  021236  004037  043226                  JSR     R0,@#SAVER      ;SAVE
4000  021242  001632                          RHWC                    ;FROM
4001  021244  002110                          WRFROM                  ;TO
4002  021246  000023                          19.                     ;NUMBER OF REGISTERS SAVED
4003
4004
4005                                          ;*AS UPPER BYTE OF RHAS CAN BE CHANGING IN A DUAL PORT
4006                                          ;*OPERATION THE UPPER BYTE OF RHAS WILL BE SAVED AS IS
4007                                          ;*SO THAT THE COMPARES ARE ONLY VALID FOR THE LOWER BYTE
4008  021250  113737  003201  002135          MOVB    @#REINTO+25,@#WRFROM+25;SAVE UPPER RHAS
4009
4010
4011                                          ;*COMPARE REGISTERS BEFORE COMMAND
4012                                          ;*WITH CONTENTS AFTER GO IS GIVEN
4013
4014  021256  004037  043430                  JSR     R0,@#COMPAR     ;COMPARE
4015  021262  003154                          REINTO                  ;GOOD BUFFER
4016  021264  002110                          WRFROM                  ;TEST BUFFER
4017  021266  000023                          19.                     ;NUMBER
4018  021270  021276                          5$                      ;RETURN FOR ERROR
4019  021272  021276                          5$                      ;SAME
4020  021274  021316                          6$                      ;RETURN FOR GOOD COMPARISON
4021  021276  013705  047320          5$:     MOV     @#ERWORD,R5     ;GETTING READY TO INDEX
4022  021302  060505                          ADD     R5,R5           ;DOUBLE ERROR WORD
4023  021304  016537  001630  042204          MOV     RHWC-2(R5),@#REGADR ;FAILING REGISTER ADDRESS
4024  021312  104001                          ERROR   1               ;IMPROPER REGISTER CHANGE
4025                                                                  ;AFTER COMMAND
4026                                                                  ;WITH GO IS GIVEN
4027  021314  000207                          RTS     PC              ;RETURN TO COMPARISON
4028
4029                                          ;*NOW GIVE INIT AND GET GO AND PIP DOWN
4030
4031  021316  052712  000040          6$:     BIS     #CLR,@R2        ;RH INITILIZE
4032  021322  013712  001774                  MOV     @#UNIT,@R2      ;REINSTATE UNIT NUMBER
4033  021326  012777  000001  160324          MOV     #DMD,@RHMR      ;SET DIAGNOSTIC MODE BIT
4034                                                                  ;THIS ENABLES COMMANDS WITHOUT MOL
4035                                                                  ;AND HOLDS RHLA FROM MOVING
4036
4037                                          ;*CHANGE REGISTERS TO EXPECTED VALUE
4038
```

```
4039  021334  042737  000001  003162          BIC     #GO,@#REINTO+6  ;SAVED RHCS1
4040  021342  042737  020000  003204          BIC     #PIP,@#REINTO+30 ;SAVED RHDS1
4041  021350  052737  000200  003204          BIS     #DRY,@#REINTO+30 ;SAVED RHDS1
4042  021356  017737  160312  003216          MOV     @RHLA,@#REINTO+42;SAVED RHLA
4043  021364  005037  003220                  CLR     @#REINTO+44     ;SAVED RHCC
4044
4045
4046                                          ;*AFTER INITIALIZE SAVE REGISTERS SO THAT
4047                                          ;*COMPARES CAN BE DONE
4048
4049  021370  004037  043226                  JSR     R0,@#SAVER      ;SAVE
4050  021374  001632                          RHWC                    ;FROM
4051  021376  002110                          WRFROM                  ;TO
4052  021400  000023                          19.                     ;NUMBER OF REGISTERS SAVED
4053
4054                                          ;*AS UPPER BYTE OF RHAS CAN BE CHANGING IN A DUAL PORT
4055                                          ;*OPERATION THE UPPER BYTE OF RHAS WILL BE SAVED AS IS
4056                                          ;*SO THAT THE COMPARES ARE ONLY VALID FOR THE LOWER BYTE
4057  021402  113737  003201  002135          MOVB    @#REINTO+25,@#WRFROM+25;SAVE UPPER RHAS
4058
4059
4060                                          ;*COMPARE REGISTERS AFTER INITALIZE
4061
4062  021410  004037  043430                  JSR     R0,@#COMPAR     ;COMPARE
4063  021414  003154                          REINTO                  ;GOOD BUFFER
4064  021416  002110                          WRFROM                  ;TEST BUFFER
4065  021420  000023                          19.                     ;NUMBER OF REGISTERS TO BE
4066                                                                  ;COMPARED
4067  021422  021430                          7$                      ;RETURN POINT FOR ERROR
4068  021424  021430                          7$                      ;SAME
4069  021426  021450                          8$                      ;RETURN POINT FOR GOOD COMPARISON
4070
4071  021430  013705  047320          7$:     MOV     @#ERWORD,R5     ;GETTING READY TO INDEX
4072  021434  060505                          ADD     R5,R5           ;DOUBLE ERROR WORD
4073  021436  016537  001630  042204          MOV     RHWC-2(R5),@#REGADR ;FAILING REGISTER ADDRESS
4074  021444  104001                          ERROR   1               ;# CONTENTS AFTER GIVING AN
4075                                                                  ;  COMMAND
4076  021446  000207                          RTS     PC              ;RETURN TO COMPARISON
4077  021450                          8$:                             ;GOOD COMPARISON
```

```
4078
4079
4080                                    ;;*****************************************************************
4081                                    ;*TEST 46        UNLOAD COMMAND TEST
4082
4083                                    ;**     THE UNLOAD COMMAND WILL BE LOADED INTO RHCS1 WITH GO
4084                                    ;**     THEN ALL REGISTERS WILL BE CHECKED
4085                                    ;**     RH CLEAR WILL BE GIVEN
4086
4087                                    ;;*****************************************************************
4088  021450  000004                    TST46:  SCOPE
4089  021452  012706  001000                    MOV     #STACK,SP       ;RESET STACK
4090  021456  012737  000046  002032            MOV     #TTNO,@#TSTNM   ;THIS SAVES TEST NUMBER
4091  021464  004737  042534                    JSR     PC,@#CLDISK     ;INIT AND SET UP GENERAL REG.
4092                                                                    ;AND UNIT NUMBER
4093  021470  012777  000001  160162            MOV     #DMD,@RHMR      ;SET DIAGNOSTIC MODE BIT
4094                                                                    ;THIS ENABLES COMMANDS WITHOUT MOL
4095                                                                    ;AND HOLDS RHLA FROM MOVING
4096
4097  021476  013777  002044  160134            MOV     @#UNLOAD,@RHCS1 ;LOAD UNLOAD COMMAND INTO RH
4098
4099                                    ;*SAVE REGISTERS FOR COMPARISON AFTER GO
4100
4101  021504  004037  043226                    JSR     R0,@#SAVER      ;SAVE
4102  021510  001632                            RHWC                    ;FROM
4103  021512  003154                            REINTO                  ;TO
4104  021514  000023                            19.                     ;NUMBER OF REGISTERS SAVED
4105
4106                                    ;*GIVE GO TO UNLOAD COMMAND
4107  021516  052777  000001  160114            BIS     #GO,@RHCS1      ;GO TO UNLOAD COMMAND
4108
4109                                    ;*CHANGE SAVED REGISTERS TO EXPECTED VALUES
4110  021524  052737  000001  003162            BIS     #GO,@#REINTO+6  ;SAVED RHCS1
4111  021532  052737  020000  003204            BIS     #PIP,@#REINTO+30 ;SAVED RHDS1
4112  021540  042737  000200  003204            BIC     #DRY,@#REINTO+30 ;SAVED RHDS1
4113
4114  021546  005737  001100                    TST     @#$PASS         ;IS THIS FIRST PASS
4115  021552  001053                            BNE     5$              ;BRANCH IF NOT FIRST PASS
4116  021554  032777  020000  157356            BIT     #SW13,@SWR      ;INHIBIT ERROR PRINT HIGH?
4117  021562  001047                            BNE     5$              ;BRANCH IF SW13 HIGH
4118
4119  021564  104401  021572                    TYPE    ,65$            ;;TYPE ASCIZ STRING
4120  021570  000441                            BR      .  64$          ;;GET OVER THE ASCIZ
4121                                  ;;65$:    .ASCIZ  <15><12>/IF DRIVE CONNECTED "STAND BY" LAMP SHOULD BE LIT ON DRIVE NO  /
4122  021674                          64$:
4123
4124  021674  013746  001774                    MOV     @#UNIT,-(SP)    ;UNIT UNDER TEST
4125  021700  104405                            TYPDS
4126
4127                                    ;*AFTER GO HAS BEEN GIVEN TO UNLOAD COMMAND
4128                                    ;*SAVED REGISTERS AGAIN SO THAT COMPARISONS CAN
4129                                    ;*BE DONE
4130
4131  021702  004037  043226           5$:      JSR     R0,@#SAVER      ;SAVE
4132  021706  001632                            RHWC                    ;FROM
4133  021710  002110                            WRFROM                  ;TO
```

```
4134  021712  000023                          19.                     ;NUMBER OF REGISTERS SAVED
4135
4136                                          ;*AS UPPER BYTE OF RHAS CAN BE CHANGING IN A DUAL PORT
4137                                          ;*OPERATION THE UPPER BYTE OF RHAS WILL BE SAVED AS IS
4138                                          ;*SO THAT THE COMPARES ARE ONLY VALID FOR THE LOWER BYTE
4139  021714  113737  003201  002135          MOVB    @#REINTO+25,@#WRFROM+25;SAVE UPPER RHAS
4140
4141
4142                                          ;*COMPARE REGISTERS BEFORE UNLOAD COMMAND
4143                                          ;*WITH AFTER GO
4144
4145  021722  004037  043430                  JSR     R0,@#COMPAR     ;COMPARE
4146  021726  003154                          REINTO                  ;GOOD BUFFER
4147  021730  002110                          WRFROM                  ;TEST BUFFER
4148  021732  000023                          19.                     ;NUMBER
4149  021734  021742                          1$                      ;RETURN FOR ERROR
4150  021736  021742                          1$                      ;SAME
4151  021740  021762                          2$                      ;RETURN FOR GOOD COMPARISON
4152  021742  013705  047320          1$:     MOV     @#ERWORD,R5     ;GETTING READY TO INDEX
4153  021746  060505                          ADD     R5,R5           ;DOUBLE ERROR WORD
4154  021750  016537  001630  042204          MOV     RHWC-2(R5),@#REGADR ;FAILING REGISTER ADDRESS
4155
4156  021756  104001                          ERROR   1               ;IMPROPER REGISTER CHANGE
4157                                                                  ;AFTER UNLOAD COMMAND
4158                                                                  ;WITH GO IS GIVEN
4159  021760  000207                          RTS     PC              ;RETURN TO COMPARISON
4160
4161                                          ;*NOW GIVE INIT AND GET ALL GO AND PIP DOWN
4162  021762  052712  000040          2$:     BIS     #CLR,@R2        ;RH INITILIZE
4163  021766  013712  001774                  MOV     @#UNIT,@R2      ;REINSTATE UNIT NUMBER
4164  021772  012777  000001  157660          MOV     #DMD,@RHMR      ;SET DIAGNOSTIC MODE BIT
4165                                                                  ;THIS ENABLES COMMANDS WITHOUT MOL
4166                                                                  ;AND HOLDS RHLA FROM MOVING
4167
4168                                          ;*CHANGE REGISTERS TO EXPECTED VALUE
4169  022000  042737  000001  003162          BIC     #GO,@#REINTO+6  ;SAVED RHCS1
4170  022006  042737  020000  003204          BIC     #PIP,@#REINTO+30 ;SAVED RHDS1
4171  022014  052737  000200  003204          BIS     #DRY,@#REINTO+30 ;SAVED RHDS1
4172  022022  017737  157646  003216          MOV     @RHLA,@#REINTO+42;SAVED RHLA
4173
4174                                          ;*AFTER INITIALIZE SAVE REGISTERS SO THAT
4175                                          ;*COMPARES CAN BE DONE
4176
4177  022030  004037  043226                  JSR     R0,@#SAVER      ;SAVE
4178  022034  001632                          RHWC                    ;FROM
4179  022036  002110                          WRFROM                  ;TO
4180  022040  000023                          19.                     ;NUMBER OF REGISTERS SAVED
4181
4182                                          ;*COMPARE REGISTERS AFTER INITIALIZE
4183                                          ;*AS UPPER BYTE OF RHAS CAN BE CHANGING IN A DUAL PORT
4184                                          ;*OPERATION THE UPPER BYTE OF RHAS WILL BE SAVED AS IS
4185                                          ;*SO THAT THE COMPARES ARE ONLY VALID FOR THE LOWER BYTE
4186  022042  113737  003201  002135          MOVB    @#REINTO+25,@#WRFROM+25;SAVE UPPER RHAS
4187
4188
4189  022050  004037  043430                  JSR     R0,@#COMPAR     ;COMPARE
```

```
4190  022054  003154                        RFINTO                  ;GOOD BUFFER
4191  022056  002110                        WRFROM                  ;TEST BUFFER
4192  022060  000023                        19.                     ;NUMBER OF REGISTERS TO BE
4193                                                                ;COMPARED
4194  022062  022070                        3$                      ;RETURN POINT FOR ERROR
4195  022064  022070                        3$                      ;SAME
4196  022066  022110                        4$                      ;RETURN POINT FOR GOOD COMPARISON
4197
4198  022070  013705  047320        3$:     MOV     @#ERWORD,R5     ;GETTING READY TO INDEX
4199  022074  060505                        ADD     R5,R5           ;DOUBLE ERROR WORD
4200  022076  016537  001630  042704        MOV     RHWC-2(R5),@#REGADR ;FAILING REGISTER ADDRESS
4201  022104  104001                        ERROR   1               ;IMPROPER REGISTER
4202                                                                ;CONTENTS AFTER GIVING AN
4203                                                                ;UNLOAD COMMAND
4204  022106  000207                        RTS     PC              ;RETURN TO COMPARISON
4205  022110                        4$:                             ;GOOD COMPARISON
4206
```

```
4207
4208  022110  032777  000020  157022          BIT     #SW4,@SWR       ;TEST FOR NO OFFSET OR RTC
4209  022116  001402                          BEQ     6$              ;IF = 0, DO THE NEXT TWO TESTS
4210  022120  000137  023076                  JMP     TST51           ;SKIP THE NEXT TWO TESTS ----------)
4211  022124                          6$:                             ;CONTINUE WITH NEXT TWO TESTS
4212
4213
4214
4215                                  ;;**********************************************************************
4216                                  ;*TEST 47        OFFSET COMMAND TEST
4217
4218                                  ;**     THE OFFSET COMMAND WILL BE LOADED INTO RHCS1 WITH GO
4219                                  ;**     THEN ALL REGISTERS WILL BE CHECKED
4220                                  ;**     RH CLEAR WILL BE GIVEN
4221                                  ;**     THEN ALL REGISTERS WILL BE CHECKED
4222
4223                                  ;;**********************************************************************
4224  022124  000004                  TST47:  SCOPE
4225  022126  012706  001000                  MOV     #STACK,SP       ;RESET STACK
4226  022132  012737  000047  002032          MOV     #TTNO,@#TSTNM   ;THIS SAVES TEST NUMBER
4227  022140  004737  042534                  JSR     PC,@#CLDISK     ;INIT AND SET UP GENERAL REG.
4228                                                                  ;AND UNIT NUMBER
4229  022144  012777  000001  157506          MOV     #DMD,@RHMR      ;SET DIAGNOSTIC MODE BIT
4230                                                                  ;THIS ENABLES COMMANDS WITHOUT MOL
4231                                                                  ;AND HOLDS RHLA FROM MOVING
4232
4233                                  ;*GIVE ONE INDEX PULSE TO CLEAR RHLA BEFORE THE START OF THIS TEST
4234  022152  052777  000004  157500          BIS     #MINX,@RHMR     ;SET INDEX PULSE
4235  022160  042777  000004  157472          BIC     #MINX,@RHMR     ;CLEAR INDEX
4236
4237                                  ;*TO ENABLE LOOP ON THIS TEST THE POSITIONER HAS TO
4238                                  ;*BE BROUGHT TO CENTER LINE
4239
4240  022166  017777  157504  157456          MOV     @RHCC,@RHCA     ;SET DESIRED CYLINDER TO RHCC
4241  022174  013711  002072                  MOV     @#SEECOM,@R1    ;SEEK COMMAND TO RHCS1
4242  022200  005211                          INC     @R1             ;GO TO SEEK COMMAND
4243
4244                                  ;*FOUR SECTOR CLOCKS ARE GIVEN TO TAKE POSITIONER OFF OFFSET POSITION
4245  022202  012700  000004                  MOV     #4,R0           ;COUNTER
4246  022206  012777  000011  157444  5$:     MOV     #MSTCK!DMD,@RHMR        ;SET SECTOR CLOCK
4247  022214  012777  000001  157436          MOV     #DMD,@RHMR      ;RESET SECTOR CLOCK
4248  022222  005300                          DEC     R0              ;COUNT
4249  022224  001370                          BNE     5$              ;BRANCH IF NOT COMPLETE
4250
4251  022226  004737  042534                  JSR     PC,@#CLDISK     ;INIT AND SET UP GENERAL REG.
4252                                                                  ;AND UNIT NUMBER
4253  022232  012777  000001  157420          MOV     #DMD,@RHMR      ;SET DIAGNOSTIC MODE BIT
4254                                                                  ;THIS ENABLES COMMANDS WITHOUT MOL
4255                                                                  ;AND HOLDS RHLA FROM MOVING
4256
4257  022240  013777  002074  157372          MOV     @#OFSETC,@RHCS1 ;LOAD AN OFFSET BIT
4258  022246  012777  000001  157374          MOV     #OF25,@RHOF     ;SET AN OFFSET BIT
4259
4260                                  ;*SAVE REGISTERS FOR COMPARISON AFTER GO
4261  022254  004037  043226                  JSR     R0,@#SAVER      ;SAVE
4262  022260  001632                          RHWC                    ;FROM
```

```
4263  022262  003154                          REINTO                  ;TO
4264  022264  000023                          19.                     ;NUMBER OF REGISTERS SAVED
4265
4266                                          ;*GIVE GO TO OFFSET COMMAND
4267  022266  052777  000001  157344          BIS     #GO,@RHCS1      ;GO TO OFFSET COMMAND
4268
4269                                          ;*CHANGE SAVED REGISTERS TO EXPECTED VALUES
4270  022274  052737  000001  003162          BIS     #GO,@#REINTO+6  ;SAVED RHCS1
4271  022302  052737  020000  003204          BIS     #PIP,@#REINTO+30 ;SAVED RHDS1
4272  022310  042737  000200  003204          BIC     #DRY,@#REINTO+30 ;SAVED RHDS1
4273
4274                                          ;*AFTER GO HAS BEEN GIVEN TO OFFSET COMMAND
4275                                          ;*SAVE REGISTERS AGAIN SO THAT COMPARISONS CAN
4276                                          ;*BE DONE
4277
4278  022316  004037  043226                  JSR     R0,@#SAVER      ;SAVE
4279  022322  001632                          RHWC                    ;FROM
4280  022324  002110                          WRFROM                  ;TO
4281  022326  000023                          19.                     ;NUMBER OF REGISTERS SAVED
4282
4283                                          ;*AS UPPER BYTE OF RHAS CAN BE CHANGING IN A DUAL PORT
4284                                          ;*OPERATION THE UPPER BYTE OF RHAS WILL BE SAVED AS IS
4285                                          ;*SO THAT THE COMPARES ARE ONLY VALID FOR THE LOWER BYTE
4286  022330  113737  003201  002135          MOVB    @#REINTO+25,@#WRFROM+25;SAVE UPPER RHAS
4287
4288
4289                                          ;*COMPARE REGISTERS BEFORE OFFSET COMMAND
4290                                          ;*WITH AFTER GO
4291
4292  022336  004037  043430                  JSR     R0,@#COMPAR     ;COMPARE
4293  022342  003154                          REINTO                  ;GOOD BUFFER
4294  022344  002110                          WRFROM                  ;TEST BUFFER
4295  022346  000023                          19.                     ;NUMBER
4296  022350  022356                          1$                      ;RETURN FOR ERROR
4297  022352  022356                          1$                      ;SAME
4298  022354  022376                          2$                      ;RETURN FOR GOOD COMPARISON
4299
4300  022356  013705  047320          1$:     MOV     @#ERWORD,R5     ;GETTING READY TO INDEX
4301  022362  060505                          ADD     R5,R5           ;DOUBLE ERROR WORD
4302  022364  016537  001630  042204          MOV     RHWC-2(R5),@#REGADR ;FAILING REGISTER ADDRESS
4303  022372  104001                          ERROR   1               ;IMPROPER REGISTER CHANGE
4304                                                                  ;AFTER OFFSET COMMAND
4305                                                                  ;WITH GO IS GIVEN
4306  022374  000207                          RTS     PC              ;RETURN TO COMPARISON
4307
4308                                          ;*NOW GIVE INIT AND GET ALL GO AND PIP DOWN
4309
4310  022376  052712  000040          2$:     BIS     #CLR,@R2        ;RH INITILIZE
4311  022402  013712  001774                  MOV     @#UNIT,@R2      ;REINSTATE UNIT NUMBER
4312  022406  012777  000001  157244          MOV     #DMD,@RHMR      ;SET DIAGNOSTIC MODE BIT
4313                                                                  ;THIS ENABLES COMMANDS WITHOUT MOL
4314                                                                  ;AND HOLDS RHLA FROM MOVING
4315
4316                                          ;*CHANGE REGISTERS TO EXPECTED VALUE
4317  022414  042737  000001  003162          BIC     #GO,@#REINTO+6  ;SAVED RHCS1
4318  022422  042737  000001  003172          BIC     #OF25,@#REINTO+16;SAVED RHOF
```

```
 4319  022430  042737  020000  003204          BIC     #PIP,@#REINTO+30 ;SAVED RHDS1
 4320  022436  052737  000200  003204          BIS     #DRY,@#REINTO+30 ;SAVED RHDS1
 4321  022444  017737  157224  003216          MOV     @RHLA,@#REINTO+42;SAVED RHLA
 4322
 4323                                          ;*AFTER INITIALIZE SAVE REGISTERS SO THAT
 4324                                          ;*COMPARES CAN BE DONE
 4325
 4326  022452  004037  043226                  JSR     R0,@#SAVER      ;SAVE
 4327  022456  001632                          RHWC                    ;FROM
 4328  022460  002110                          WRFROM                  ;TO
 4329  022462  000023                          19.                     ;NUMBER OF REGISTERS SAVED
 4330
 4331                                          ;*AS UPPER BYTE OF RHAS CAN BE CHANGING IN A DUAL PORT
 4332                                          ;*OPERATION THE UPPER BYTE OF RHAS WILL BE SAVED AS IS
 4333                                          ;*SO THAT THE COMPARES ARE ONLY VALID FOR THE LOWER BYTE
 4334  022464  113737  003201  002135          MOVB    @#REINTO+25,@#WRFROM+25;SAVE UPPER RHAS
 4335
 4336                                          ;*COMPARE REGISTERS AFTER INITIALIZE
 4337  022472  004037  043430                  JSR     R0,@#COMPAR     ;COMPARE
 4338  022476  003154                          REINTO                  ;GOOD BUFFER
 4339  022500  002110                          WRFROM                  ;TEST BUFFER
 4340  022502  000023                          19.                     ;NUMBER OF REGISTERS TO BE
 4341                                                                  ;COMPARED
 4342  022504  022512                          3$                      ;RETURN POINT FOR ERROR
 4343  022506  022512                          3$                      ;SAME
 4344  022510  022532                          4$                      ;RETURN POINT FOR GOOD COMPARISON
 4345
 4346  022512  013705  047320          3$:     MOV     @#ERWORD,R5     ;GETTING READY TO INDEX
 4347  022516  060505                          ADD     R5,R5           ;DOUBLE ERROR WORD
 4348  022520  016537  001630  042204          MOV     RHWC-2(R5),@#REGADR ;FAILING REGISTER ADDRESS
 4349  022526  104001                          ERROR   1               ;IMPROPER REGISTER
 4350                                                                  ;CONTENTS AFTER GIVING AN
 4351                                                                  ;INITIALIZE FOLLOWING A
 4352                                                                  ;OFFSET COMMAND
 4353  022530  000207                          RTS     PC              ;RETURN TO COMPARISON
 4354
 4355  022532                          4$:                             ;GOOD COMPARISON
```

```
4356
4357
4358                                    ;;*****************************************************************
4359                                    ;*TEST 50       RETURN TO CENTER LINE COMMAND TEST
4360
4361                                    ;**     THE RETURN TO CENTER LINE COMMAND WILL BE LOADED INTO RHCS1 WITH GO
4362                                    ;**     THEN ALL REGISTERS WILL BE CHECKED
4363                                    ;**     RH CLEAR WILL BE GIVEN
4364                                    ;**     THEN ALL REGISTERS WILL BE CHECKED
4365
4366                                    ;;*****************************************************************
4367  022532  000004                    TST50:  SCOPE
4368  022534  012706  001000                    MOV     #STACK,SP       ;RESET STACK
4369  022540  012737  000050  002032            MOV     #TTNO,@#TSTNM   ;THIS SAVES TEST NUMBER
4370  022546  004737  042534                    JSR     PC,@#CLDISK     ;INIT AND SET UP GENERAL REG.
4371                                                                    ;AND UNIT NUMBER
4372  022552  012777  000001  157100            MOV     #DMD,@RHMR      ;SET DIAGNOSTIC MODE BIT
4373                                                                    ;THIS ENABLES COMMANDS WITHOUT MOL
4374                                                                    ;AND HOLDS RHLA FROM MOVING
4375
4376                                    ;*GIVE ONE INDEX PULSE TO CLEAR RHLA BEFORE THE START OF THIS TEST
4377  022560  052777  000004  157072            BIS     #MINX,@RHMR     ;SET INDEX PULSE
4378  022566  042777  000004  157064            BIC     #MINX,@RHMR     ;CLEAR INDEX
4379
4380
4381  022574  013777  002076  157036            MOV     @#RETCL,@RHCS1  ;LOAD RETURN TO CENTER LINE COMMAND INTO RHCS1
4382
4383                                    ;*SAVE REGISTERS FOR COMPARISON AFTER GO
4384  022602  004037  043226                    JSR     R0,@#SAVER      ;SAVE
4385  022606  001632                            RHWC                    ;FROM
4386  022610  003154                            REINTO                  ;TO
4387  022612  000023                            19.                     ;NUMBER OF REGISTERS SAVED
4388
4389                                    ;*GIVE GO TO RETURN TO CENTER LINE COMMAND
4390  022614  052777  000001  157016            BIS     #GO,@RHCS1      ;GO TO RETURN TO CENTER COMMAND
4391
4392
4393                                    ;*FOUR SECTOR CLOCKS ARE GIVEN TO TAKE POSITIONER TO CENTER LINE
4394  022622  012700  000004                    MOV     #4,R0           ;COUNTER
4395  022626  012777  000011  157024  5$:       MOV     #MSTCK!DMD,@RHMR;SET SECTOR CLOCK
4396  022634  012777  000001  157016            MOV     #DMD,@RHMR      ;RESET SECTOR CLOCK
4397  022642  005300                            DEC     R0              ;COUNT
4398  022644  001370                            BNE     5$              ;BRANCH IF NOT COMPLETE
4399
4400
4401                                    ;*CHANGE SAVED REGISTERS TO EXPECTED VALUES
4402  022646  052737  000001  003162            BIS     #GO,@#REINTO+6  ;SAVED RHCS1
4403  022654  052737  020000  003204            BIS     #PIP,@#REINTO+30 ;SAVED RHDS1
4404  022662  042737  000200  003204            BIC     #DRY,@#REINTO+30 ;SAVED RHDS1
4405
4406                                    ;*AFTER GO HAS BEEN GIVEN TO RETURN TO CENTER LINE COMMAND
4407                                    ;*SAVE REGISTERS AGAIN SO THAT COMPARISONS CAN
4408                                    ;*BE DONE
4409  022670  004037  043226                    JSR     R0,@#SAVER      ;SAVE
4410  022674  001632                            RHWC                    ;FROM
4411  022676  002110                            WRFROM                  ;TO
```

```
4412  022700  000023                        19.                     ;NUMBER OF REGISTERS SAVED
4413
4414                                        ;*AS UPPER BYTE OF RHAS CAN BE CHANGING IN A DUAL PORT
4415                                        ;*OPERATION THE UPPER BYTE OF RHAS WILL BE SAVED AS IS
4416                                        ;*SO THAT THE COMPARES ARE ONLY VALID FOR THE LOWER BYTE
4417  022702  113737  003201  002135        MOVB    @#REINTO+25,@#WRFROM+25;SAVE UPPER RHAS
4418
4419
4420                                        ;*COMPARE REGISTERS BEFORE RETURN TO CENTER LINE COMMAND
4421                                        ;*WITH AFTER GO
4422  022710  004037  043430                JSR     R0,@#COMPAR     ;COMPARE
4423  022714  003154                        REINTO                  ;GOOD BUFFER
4424  022716  002110                        WRFROM                  ;TEST BUFFER
4425  022720  000023                        19.                     ;NUMBER
4426  022722  022730                        1$                      ;RETURN FOR ERROR
4427  022724  022730                        1$                      ;SAME
4428  022726  022750                        2$                      ;RETURN FOR GOOD COMPARISON
4429
4430  022730  013705  047320        1$:     MOV     @#ERWORD,R5     ;GETTING READY TO INDEX
4431  022734  060505                        ADD     R5,R5           ;DOUBLE ERROR WORD
4432  022736  016537  001630  042204        MOV     RHWC-2(R5),@#REGADR ;FAILING REGISTER ADDRESS
4433  022744  104001                        ERROR   1               ;IMPROPER REGISTER CHANGE
4434                                                                ;AFTER RETURN TO CENTER LINE COMMAND
4435                                                                ;WITH GO IS GIVEN
4436  022746  000207                        RTS     PC              ;RETURN TO COMPARISON
4437
4438                                        ;*NOW GIVE INIT AND GET ALL GO AND PIP DOWN
4439
4440  022750  052712  000040        2$:     BIS     #CLR,@R2        ;RH INITILIZE
4441  022754  013712  001774                MOV     @#UNIT,@R2      ;REINSTATE UNIT NUMBER
4442  022760  012777  000001  156672        MOV     #DMD,@RHMR      ;SET DIAGNOSTIC MODE BIT
4443                                                                ;THIS ENABLES COMMANDS WITHOUT MOL
4444                                                                ;AND HOLDS RHLA FROM MOVING
4445
4446                                        ;*CHANGE REGISTERS TO EXPECTED VALUE
4447  022766  042737  000001  003162        BIC     #GO,@#REINTO+6  ;SAVED RHCS1
4448  022774  042737  020000  003204        BIC     #PIP,@#REINTO+30 ;SAVED RHDS1
4449  023002  052737  000200  003204        BIS     #DRY,@#REINTO+30 ;SAVED RHDS1
4450  023010  017737  156660  003216        MOV     @RHLA,@#REINTO+42;SAVED RHLA
4451
4452                                        ;*AFTER INITIALIZE SAVE REGISTERS SO THAT
4453                                        ;*COMPARES CAN BE DONE
4454  023016  004037  043226                JSR     R0,@#SAVER      ;SAVE
4455  023022  001632                        RHWC                    ;FROM
4456  023024  002110                        WRFROM                  ;TO
4457  023026  000023                        19.                     ;NUMBER OF REGISTERS SAVED
4458
4459                                        ;*AS UPPER BYTE OF RHAS CAN BE CHANGING IN A DUAL PORT
4460                                        ;*OPERATION THE UPPER BYTE OF RHAS WILL BE SAVED AS IS
4461                                        ;*SO THAT THE COMPARES ARE ONLY VALID FOR THE LOWER BYTE
4462  023030  113737  003201  002135        MOVB    @#REINTO+25,@#WRFROM+25;SAVE UPPER RHAS
4463
4464
4465                                        ;*COMPARE REGISTERS AFTER INITIALIZE
4466  023036  004037  043430                JSR     R0,@#COMPAR     ;COMPARE
4467  023042  003154                        REINTO                  ;GOOD BUFFER
```

```
 4468  023044  002110                          WRFROM                    ;TEST BUFFER
 4469  023046  000023                          19.                       ;NUMBER OF REGISTERS TO BE
 4470                                                                    ;COMPARED
 4471  023050  023056                          3$                        ;RETURN POINT FOR ERROR
 4472  023052  023056                          3$                        ;SAME
 4473  023054  023076                          4$                        ;RETURN POINT FOR GOOD COMPARISON
 4474
 4475
 4476  023056  013705  047320          3$:     MOV     @#ERWORD,R5       ;GETTING READY TO INDEX
 4477  023062  060505                          ADD     R5,R5             ;DOUBLE ERROR WORD
 4478  023064  016537  001630  042204          MOV     RHWC-2(R5),@#REGADR ;FAILING REGISTER ADDRESS
 4479  023072  104001                          ERROR   1                 ;IMPROPER REGISTER
 4480                                                                    ;CONTENTS AFTER GIVING AN
 4481                                                                    ;INITIALIZE FOLLOWING A RETURN TO
 4482                                                                    ;CENTER LINE COMMAND
 4483  023074  000207                          RTS     PC                ;RETURN TO COMPARISON
 4484
 4485  023076                          4$:                               ;GOOD COMPARISON
 4486
```

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

                                ;;*****************************************************************
023076  000004                  TST51:  SCOPE
023100  012706  001000                  MOV     #STACK,SP       ;RESET STACK
023104  012737  000051  002032          MOV     #TTNO,@#TSTNM   ;THIS SAVES TEST NUMBER
023112  004737  042534                  JSR     PC,@#CLDISK     ;INIT AND SET UP GENERAL REG.
                                                                ;AND UNIT NUMBER
023116  012777  000001  156534          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

023124  052777  000004  156526          BIS     #MINX,@RHMR     ;SET INDEX PULSE
023132  042777  000004  156520          BIC     #MINX,@RHMR     ;CLEAR INDEX

023140  013777  002046  156472          MOV     @#RECALI,@RHCS1 ;LOAD RECALIBRATE COMMAND INTO RHCS1

                                ;*SAVE REGISTERS FOR COMPARISON AFTER GO
023146  004037  043226                  JSR     R0,@#SAVER      ;SAVE
023152  001632                          RHWC                    ;FROM
023154  003154                          REINTO                  ;TO
023156  000023                          19.                     ;NUMBER OF REGISTERS SAVED

                                ;*GIVE GO TO RECALIBRATE COMMAND
023160  052777  000001  156452          BIS     #GO,@RHCS1      ;GO TO RECALIBRATE COMMAND

                                ;*FOUR SECTOR CLOCKS ARE GIVEN TO TAKE POSITIONER TO CYLINDER 0
023166  012700  000004                  MOV     #4,R0           ;COUNTER
023172  012777  000011  156460  5$:     MOV     #MSTCK!DMD,@RHMR;SET SECTOR CLOCK
023200  012777  000001  156452          MOV     #DMD,@RHMR      ;RESET SECTOR CLOCK
023206  005300                          DEC     R0              ;COUNT
023210  001370                          BNE     5$              ;BRANCH IF NOT COMPLETE

                                ;*CHANGE SAVED REGISTERS TO EXPECTED VALUES
023212  052737  000001  003162          BIS     #GO,@#REINTO+6  ;SAVED RHCS1
023220  052737  020000  003204          BIS     #PIP,@#REINTO+30 ;SAVED RHDS1
023226  042737  000200  003204          BIC     #DRY,@#REINTO+30 ;SAVED RHDS1

                                ;*AFTER GO HAS BEEN GIVEN TO RECALIBRATE COMMAND
                                ;*SAVE REGISTERS AGAIN SO THAT COMPARISONS CAN
                                ;*BE DONE
023234  004037  043226                  JSR     R0,@#SAVER      ;SAVE
023240  001632                          RHWC                    ;FROM
```

```
4543  023242  002110                          WRFROM                  ;TO
4544  023244  000023                          19.                     ;NUMBER OF REGISTERS SAVED
4545
4546                                          ;*AS UPPER BYTE OF RHAS CAN BE CHANGING IN A DUAL PORT
4547                                          ;*OPERATION THE UPPER BYTE OF RHAS WILL BE SAVED AS IS
4548                                          ;*SO THAT THE COMPARES ARE ONLY VALID FOR THE LOWER BYTE
4549  023246  113737  003201  002135          MOVB    @#REINTO+25,@#WRFROM+25;SAVE UPPER RHAS
4550
4551
4552                                          ;*COMPARE REGISTERS BEFORE RECALIBRATE COMMAND
4553                                          ;*WITH AFTER GO
4554  023254  004037  043430                  JSR     R0,@#COMPAR     ;COMPARE
4555  023260  003154                          REINTO                  ;GOOD BUFFER
4556  023262  002110                          WRFROM                  ;TEST BUFFER
4557  023264  000023                          19.                     ;NUMBER
4558  023266  023274                          1$                      ;RETURN FOR ERROR
4559  023270  023274                          1$                      ;SAME
4560  023272  023314                          2$                      ;RETURN FOR GOOD COMPARISON
4561
4562  023274  013705  047320          1$:     MOV     @#ERWORD,R5     ;GETTING READY TO INDEX
4563  023300  060505                          ADD     R5,R5           ;DOUBLE ERROR WORD
4564  023302  016537  001630  042204          MOV     RHMC-2(R5),@#REGADR ;FAILING REGISTER ADDRESS
4565  023310  104001                          ERROR   1               ;IMPROPER REGISTER CHANGE
4566                                                                  ;AFTER RECALIBRATE COMMAND
4567                                                                  ;WITH GO IS GIVEN
4568  023312  000207                          RTS     PC              ;RETURN TO COMPARISON
4569
4570                                          ;*NOW GIVE INIT AND GET ALL GO AND PIP DOWN
4571
4572  023314  052712  000040          2$:     BIS     #CLR,@R2        ;RH INITILIZE
4573  023320  013712  001774                  MOV     @#UNIT,@R2      ;REINSTATE UNIT NUMBER
4574  023324  012777  000001  156326          MOV     #DMD,@RHMR      ;SET DIAGNOSTIC MODE BIT
4575                                                                  ;THIS ENABLES COMMANDS WITHOUT MOL
4576                                                                  ;AND HOLDS RHLA FROM MOVING
4577
4578                                          ;*CHANGE REGISTERS TO EXPECTED VALUE
4579  023332  042737  000001  003162          BIC     #GO,@#REINTO+6  ;SAVED RHCS1
4580  023340  042737  020000  003204          BIC     #PIP,@#REINTO+30 ;SAVED RHDS1
4581  023346  052737  000200  003204          BIS     #DRY,@#REINTO+30 ;SAVED RHDS1
4582  023354  017737  156314  003216          MOV     @RHLA,@#REINTO+42;SAVED RHLA
4583
4584                                          ;*AFTER INITIALIZE SAVE REGISTERS SO THAT
4585                                          ;*COMPARES CAN BE DONE
4586  023362  004037  043226                  JSR     R0,@#SAVER      ;SAVE
4587  023366  001632                          RHWC                    ;FROM
4588  023370  002110                          WRFROM                  ;TO
4589  023372  000023                          19.                     ;NUMBER OF REGISTERS SAVED
4590
4591                                          ;*AS UPPER BYTE OF RHAS CAN BE CHANGING IN A DUAL PORT
4592                                          ;*OPERATION THE UPPER BYTE OF RHAS WILL BE SAVED AS IS
4593                                          ;*SO THAT THE COMPARES ARE ONLY VALID FOR THE LOWER BYTE
4594  023374  113737  003201  002135          MOVB    @#REINTO+25,@#WRFROM+25;SAVE UPPER RHAS
4595
4596
4597                                          ;*COMPARE REGISTERS AFTER INITIALIZE
4598  023402  004037  043430                  JSR     R0,@#COMPAR     ;COMPARE
```

```
4599  023406  003154                          REINTO                          ;GOOD BUFFER
4600  023410  002110                          WRFROM                          ;TEST BUFFER
4601  023412  000023                          19.                             ;NUMBER OF REGISTERS TO BE
4602                                                                          ;COMPARED
4603  023414  023422                          3$                              ;RETURN POINT FOR ERROR
4604  023416  023422                          3$                              ;SAME
4605  023420  023442                          4$                              ;RETURN POINT FOR GOOD COMPARISON
4606
4607  023422  013705  047320          3$:     MOV     @#ERWORD,R5             ;GETTING READY TO INDEX
4608  023426  060505                          ADD     R5,R5                   ;DOUBLE ERROR WORD
4609  023430  016537  001630  042204          MOV     RHWC-2(R5),@#REGADR ;FAILING REGISTER ADDRESS
4610  023436  104001                          ERROR   1                       ;IMPROPER REGISTER
4611                                                                          ;CONTENTS AFTER GIVING AN
4612                                                                          ;INITIALIZE FOLLOWING A
4613                                                                          ;RECALIBRATE COMMAND
4614  023440  000207                          RTS     PC                      ;RETURN TO COMPARISON
4615
4616  023442                          4$:                                     ;GOOD COMPARISON
```

```
4617
4618
4619                                    ;;*****************************************************************
4620                                    ;*TEST 52        RELEASE COMMAND TEST
4621
4622                                    ;**     THE RELEASE COMMAND WILL BE LOADED INTO RHCS1 WITH GO
4623                                    ;**     THEN ALL REGISTERS WILL BE CHECKED
4624                                    ;**     RH CLEAR WILL BE GIVEN
4625                                    ;**     THEN ALL REGISTERS WILL BE CHECKED
4626
4627                                    ;;*****************************************************************
4628  023442  000004                    TST52:  SCOPE
4629  023444  012706  001000                    MOV     #STACK,SP       ;RESET STACK
4630  023450  012737  000052  002032            MOV     #TTNO,@#TSTNM   ;THIS SAVES TEST NUMBER
4631  023456  004737  042534                    JSR     PC,@#CLDISK     ;INIT AND SET UP GENERAL REG.
4632                                                                    ;AND UNIT NUMBER
4633  023462  012777  000001  156170            MOV     #DMD,@RHMR      ;SET DIAGNOSTIC MODE BIT
4634                                                                    ;THIS ENABLES COMMANDS WITHOUT MOL
4635                                                                    ;AND HOLDS RHLA FROM MOVING
4636
4637  023470  013777  002052  156142            MOV     @#RELEASE,@RHCS1 ;LOAD RELEASE COMMAND INTO RHCS1
4638
4639                                    ;*SAVE REGISTERS FOR COMPARISON AFTER GO
4640  023476  004037  043226                    JSR     R0,@#SAVER      ;SAVE
4641  023502  001632                    RHWC                            ;FROM
4642  023504  003154                    REINTO                          ;TO
4643  023506  000023                    19.                             ;NUMBER OF REGISTERS SAVED
4644
4645                                    ;*GIVE GO TO RELEASE COMMAND
4646  023510  052777  000001  156122    BIS     #GO,@RHCS1      ;GO TO RELEASE COMMAND
4647  023516  052777  000001  156134    BIS     #DMD,@RHMR      ;SET DMD TO HOLD RHLA
4648
4649                                    ;*CHANGE SAVED REGISTERS TO EXPECTED VALUES
4650                                    ;*AFTER GO HAS BEEN GIVEN TO RELEASE COMMAND
4651  023524  052737  000001  003202    BIS     #DMD,@#REINTO+26;SAVED RHMR
4652  023532  017737  156136  003216    MOV     @RHLA,@#REINTO+42;SAVED RHLA
4653
4654
4655                                    ;*SAVE REGISTERS AGAIN SO THAT COMPARISONS CAN
4656                                    ;*BE DONE
4657  023540  004037  043226            JSR     R0,@#SAVER      ;SAVE
4658  023544  001632                    RHWC                    ;FROM
4659  023546  002110                    WRFROM                  ;TO
4660  023550  000023                    19.                     ;NUMBER OF REGISTERS SAVED
4661
4662                                    ;*AS UPPER BYTE OF RHAS CAN BE CHANGING IN A DUAL PORT
4663                                    ;*OPERATION THE UPPER BYTE OF RHAS WILL BE SAVED AS IS
4664                                    ;*SO THAT THE COMPARES ARE ONLY VALID FOR THE LOWER BYTE
4665  023552  113737  003201  002135    MOVB    @#REINTO+25,@#WRFROM+25;SAVE UPPER RHAS
4666
4667
4668                                    ;*COMPARE REGISTERS BEFORE RELEASE COMMAND
4669                                    ;*WITH AFTER GO
4670  023560  004037  043430            JSR     R0,@#COMPAR     ;COMPARE
4671  023564  003154                    REINTO                  ;GOOD BUFFER
4672  023566  002110                    WRFROM                  ;TEST BUFFER
```

K9

```
 4673  023570  000023                          19.                     ;NUMBER
 4674  023572  023600                          1$                      ;RETURN FOR ERROR
 4675  023574  023600                          1$                      ;SAME
 4676  023576  023620                          2$                      ;RETURN FOR GOOD COMPARISON
 4677
 4678  023600  013705  047320          1$:     MOV     @#ERWORD,R5     ;GETTING READY TO INDEX
 4679  023604  060505                          ADD     R5,R5           ;DOUBLE ERROR WORD
 4680  023606  016537  001630  042204          MOV     RHWC-2(R5),@#REGADR ;FAILING REGISTER ADDRESS
 4681  023614  104001                          ERROR   1               ;IMPROPER REGISTER CHANGE
 4682                                                                  ;AFTER RELEASE COMMAND
 4683                                                                  ;WITH GO IS GIVEN
 4684  023616  000207                          RTS     PC              ;RETURN TO COMPARISON
 4685  023620                          2$:
 4686
```

```
                                ;;*****************************************************************
                                ;*TEST 53        MAKE CURRENT CYLINDER = 0
                                ;;*****************************************************************
023620  000004                TST53:  SCOPE
023622  012706  001000                MOV     #STACK,SP       ;RESET STACK
023626  012737  000053  002032        MOV     #TTNO,@#TSTNM   ;THIS SAVES TEST NUMBER
023634  004737  042534                JSR     PC,@#CLDISK     ;INIT DRIVE
023640  012777  000001  156012        MOV     #DMD,@RHMR      ;SET DIAGNOSTIC MODE
023646  004037  045106                JSR     R0,@#MAKECYL    ;SUBROUTINE TO GIVE A SEEK
                                                            ;COMMAND FOLOWED BY AN INIT
                                                            ;THIS SHOULD CHANGE RHCC
023652  000000                        0                       ;CHANGE RHCC TO 0


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

                                ;;*****************************************************************
023654  000004                TST54:  SCOPE
023656  012706  001000                MOV     #STACK,SP       ;RESET STACK
023662  012737  000054  002032        MOV     #TTNO,@#TSTNM   ;THIS SAVES TEST NUMBER
023670  004737  042534                JSR     PC,@#CLDISK     ;INIT AND SET UP GENERAL REGISTERS

                                ;*THESE ARE REGULAR SET UPS FOR SEARCH COMMAND
023674  012777  000025  155742        MOV     #21.,@RHDST     ;DESIRED SECTOR/TRACK REGISTER
                                                            ;TRACK 0 SECTOR 21
023702  005077  155744                CLR     @RHCA           ;DESIRED CYLINDER =0
023706  012777  010000  155734        MOV     #FMT22,@RHOF    ;FORMAT BIT=1 (16 BITS PER WORD)
023714  013711  002054                MOV     @#SERCH,@R1     ;FILL SEARCH COMMAND IN RHCS1

                                ;*NOW SAVE REGISTERS STARTING FROM RHWC IN WRITE FROM BUFFER
023720  004037  043226                JSR     R0,@#SAVER      ;SAVE REGISTERS FOR COMPARISON
                                                            ;AT THE END OF THE SEARCH
023724  001632                        RHWC                    ;START SAVING FROM RHWC
023726  003154                        REINTO                  ;SAVE INTO REINTO
023730  000023                        19.                     ;NUMBER OF REGISTERS SAVED

023732  004737  042570                JSR     PC,@#CHECKT     ;CHECK THAT DVA,RDY,DPR,DRY = 1
023736  104401  062145                TYPE    ,CPHALT         ;AND THAT NO OTHERS = 1.  CANNOT CON-
                                                            ;TINUE TESTING IF BOTH AREN'T TRUE
023742  000000                        HALT                    ;STOP THE TEST

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

```
4743  023744  005037  001200                  CLR     @#$TMP1         ;THIS WILL HAVE THE EXPECTED
4744                                                                    ;VALUE OF RHLA REGISTER
4745
4746  023750  013700  001660                  MOV     @#RHMR,R0       ;NOW R0 HAS MAINTENANCE REG. ADDR.
4747  023754  017703  155664                  MOV     @RHDST,R3       ;GET DESIRED SECTOR/TRACK REG.
4748  023760  042703  177400                  BIC     #177400,R3      ;GET SECTOR ONLY
4749  023764  010337  053244                  MOV     R3,@#SECTR      ;DUPLICATE SECTOR
4750  023770  012710  000001                  MOV     #DMD,@R0        ;S
4751  023774  052777  000001  155636          BIS     #GO,@RHCS1      ;GO
4752  024002  052710  000010                  BIS     #MSTCK,@R0      ;SET SECTOR CLOCK
4753  024006  042710  000010                  BIC     #MSTCK,@R0      ;CLEAR SECTOR CLOCK
4754  024012  000240                          NOP                     ;ALLOW TIME BETWEEN SECTOR CLOCKS
4755  024014  052710  000010                  BIS     #MSTCK,@R0      ;SET SECTOR CLOCK
4756  024020  042710  000010                  BIC     #MSTCK,@R0      ;CLEAR SECTOR CLOCK
4757  024024  000240                          NOP                     ;ALLOW TIME BETWEEN SECTOR CLOCKS
4758  024026  052710  000014                  BIS     #MINX!MSTCK,@R0 ;SET INDEX AND SECTOR CLOCK
4759  024032  012710  000001                  MOV     #DMD,@R0        ;RESET INDEX AND SECTOR CLOCK
4760  024036  005703                          TST     R3              ;IF SECTOR REQUIRED JUMP OUT
4761  024040  001555                          BEQ     11$             ;BRANCH OF SECTOR ZERO REQUIRED
4762
4763                                          ;*AFTER THE INDEX PULSE RHLA WILL BE CHECKED TO BE ZERO
4764                                          ;*AND $TMP4 WILL BE SET UP TO COUNT BYTES
4765  024042  012737  001140  001206  1$:     MOV     #608.,@#$TMP4   ;THERE ARE 608 BYTES PER SECTOR
4766  024050  017737  155620  001126          MOV     @RHLA,@#$RDDAT  ;SAVE RHLA
4767  024056  017737  155562  001720          MOV     @RHDST,@#DST    ;SAVE DESIRED SECTOR TRACK
4768  024064  023737  001200  001126          CMP     @#$TMP1,@#$BDDAT ;RHLA SHOULD BE HAVE EXTENSION
4769                                                                  ;FIELD EQUAL TO ZERO
4770  024072  001414                          BEQ     2$              ;BRANCH IF GOOD
4771  024074  013737  053244  001202          MOV     @#SECTR,@#$TMP2 ;GET SECTOR SOUGHT
4772  024102  160337  001202                  SUB     R3,@#$TMP2      ;$TMP2 NOW HAS PRESENT SECTOR
4773  024106  012746  001140                  MOV     #608.,-(SP)     ;NUMBER OF BYTES PER SECTOR
4774  024112  163716  001206                  SUB     @#$TMP4,(SP)    ;(SP)HAS PRESENT BYTE NUMBER
4775  024116  012637  001204                  MOV     (SP)+,@#$TMP3   ;PRESENT BYTE NUMBER
4776  024122  104024                          ERROR   24              ;LOOK AHEAD REGISTER AT THE BEGINING OF A
4777                                                                  ;SECTOR IS IN ERROR
4778
4779
4780                                          ;*NOW THE 304 WORDS WILL START
4781                                          ;*FOR FIRST BYTE CLOCK WILL BE INDEPENDENT OF
4782                                          ;*SECTOR CLOCK THEN IT WILL COINCIDE FOREVER TILL
4783                                          ;*THE BEGINNING OF NEXT SECTOR
4784
4785                                          ;*ONE WORD ONLY THAT IS TWO BYTES
4786
4787  024124  012702  000010          2$:     MOV     #8.,R2          ;BYTE
4788  024130  012705  000002                  MOV     #2,R5           ;BYTES PER WORD
4789  024134  000404                          BR      4$
4790  024136  052710  000012          3$:     BIS     #MSTCK!MCLK,@R0 ;SET SECTOR AND CLOCK
4791  024142  042710  000012                  BIC     #MSTCK!MCLK,@R0 ;CLEAR SECTOR AND CLOCK
4792  024146  052710  000002          4$:     BIS     #MCLK,@R0       ;SET CLOCK
4793  024152  042710  000002                  BIC     #MCLK,@P0       ;CLEAR CLOCK
4794  024156  005302                          DEC     R2              ;BYTE COUNTER
4795  024160  001372                          BNE     4$              ;BRANCH IF BYTE NOT COMPLETE
4796  024162  005337  001206                  DEC     @#$TMP4         ;BYTE COUNT DOWN
4797  024166  012702  000007                  MOV     #7,P2           ;SETUP FOR SECOND BYTE
4798  024172  005305                          DEC     R5              ;IS WORD COMPLETE?
```

AD

```
4799  024174  001360                          BNE     3$              ;BRENCH IF NOT COMPLETE
4800                                                                  ;TO GIVE SECTOR CLOCK AND CLOCK
4801
4802                                          ;*NOW 303 WORDS ARE LEFT ALL ARE IDENTICAL
4803                                          ;*THAT IS 606 IDENTICAL BYTES WILL BE GIVEN
4804                                          ;*RHLA WILL BE CHECKED STAR TO COUNT AFTER
4805                                          ;*BEGINNING OF SECTOR PULSE
4806                                          ;*AFTER 128 BYTES (2 BYTES ARE ALREADY GIVEN)
4807                                          ;*SO 127 MORE
4808                                          ;*THEN RHLA WILL BE CHECKED AFTER 128 MORE BYTES
4809                                          ;*THEN RHLA WILL BE CHECKED AFTER 256 MORE BYTES
4810                                          ;*THEN THE TOTAL OF 608 BYTES WILL BE COMPLETED
4811                                          ;*AND RHLA WILL BE MADE READY FOR NEXT SECTOR
4812                                          ;*AND RHLA WILL BE CHECKED
4813
4814  024176  012705  000100                  MOV     #64.,R5         ;R5 WILL KEEP TRACK WHEN
4815                                                                  ;EXTENSION FIELD IS TO BE CHECKED
4816  024202  012701  000177                  MOV     #127.,R1        ;FIRST TIME CHECK EXTENSION FIELD
4817                                                                  ;AFTER 127 MORE BYTES
4818  024206  012702  000007          5$:     MOV     #7,R2           ;CLOCKS PER BYTE COUNTER
4819  024212  052710  000012                  BIS     #MSTCK!MCLK,@R0 ;SET SECTOR CLOCK AND CLOCK
4820  024216  042710  000012                  BIC     #MSTCK!MCLK,@R0 ;CLEAR SECTOR CLOCK AND CLOCK
4821  024222  052710  000002          6$:     BIS     #MCLK,@R0       ;SET CLOCK
4822  024226  042710  000002                  BIC     #MCLK,@R0       ;RESET CLOCK
4823  024232  005302                          DEC     R2              ;COUNT DOWN CLOCKS PER BYTE
4824  024234  001372                          BNE     6$              ;BRANCH IF BYTE NOT COMPLETE
4825  024236  005337  001206                  DEC     @#$TMP4         ;COUNT DOWN BYTES
4826  024242  001436                          BEQ     10$             ;BRANCHOUT IF 608 BYTES DONE
4827  024244  005301                          DEC     R1              ;COUNT DOWN NUMBER OF BYTES
4828                                                                  ;TO CHECK EXTENSION FIELD
4829  024246  001357                          BNE     5$              ;BRANCH IF EXTENSION FIELD NOT
4830                                                                  ;TO BE CHECKED YET
4831
4832                                          ;*NOW THE EXTENSION FIELD OF THE LOOK AHEAD REGISTER
4833                                          ;*WILL BE CHECKED
4834
4835  024250  062737  000020  001200          ADD     #20,@#$TMP1     ;GET TO THE NEXT EXTENSION
4836  024256  017737  155412  001126          MOV     @RHLA,@#$BDDAT  ;GET RHLA FOR COMPARISON
4837
4838  024264  017737  155354  001720          MOV     @RHDST,@#DST    ;SAVE DESIRED SECTOR TRACK
4839  024272  023737  001200  001126          CMP     @#$TMP1,@#$BDDAT ;CHECK VALUE OF RHLA
4840  024300  001414                          BEQ     7$              ;BRANCH IF GOOD
4841  024302  013737  053244  001202          MOV     @#SECTR,@#$TMP2 ;GET SECTOR SOUGHT
4842  024310  160337  001202                  SUB     R3,@#$TMP2      ;$TMP2 NOW HAS PRESENT SECTOR
4843  024314  012746  001140                  MOV     #608.,-(SP)     ;NUMBER OF BYTES PER SECTOR
4844  024320  163716  001206                  SUB     @#$TMP4,(SP)    ;(SP) HAS PRESENT BYTE NUMBER
4845  024324  012637  001204                  MOV     (SP)+,@#$TMP3   ;PRESENT BYTE NUMBER
4846  024330  104025                          ERROR   25              ;LOOK AHEAD ERROR IN THE MIDDLE
4847                                                                  ;OF A SECTOR IS IN ERROR
4848
4849  024332  060505                  7$:     ADD     R5,R5           ;GET NEXT STEP TO CHECK EXTENSION FIELD
4850  024334  010501                          MOV     R5,R1           ;PUT IN COUNTER
4851  024336  000723                          BR      5$              ;BRANCH BACK SECTOR
4852                                                                  ;IS NOT COMPLETE
4853  024340  062737  000020  001200  10$:     ADD     #20,@#$TMP1
4854  024346  052710  000010                  BIS     #MSTCK,@R0      ;THESE TWO INSTRUCTIONS GIVE
```

```
4855  024352  042710  000010                  BIC     #MSTCK,@R0      ;ONE SECTOR CLOCK EXTRA
4856  024356  000240                          NOP                     ;ALLOW TIME BETWEEN SECTOR CLOCK
4857  024360  052710  000010                  BIS     #MSTCK,@R0      ;THESE TWO INSTRUCTIONS GIVE
4858  024364  042710  000010                  BIC     #MSTCK,@R0      ;ONE SECTOR CLOCK EARLY
4859                                                                  ;BEFORE THE NEXT SECTOR
4860  024370  005303                          DEC     R3              ;IS REQUIRED NO OF SECTORS COMPLETE
4861  024372  001223                          BNE     1$              ;BRANCH IF NOT
4862
4863                                          ;*NOW THE REQUIRED SECTOR IS REACHED
4864                                          ;*ONE SECTOR CLOCK WILL BE GIVEN TO GET SECTOR PULSE
4865                                          ;*DOWN AND HENCE ATA UP
4866
4867  024374  012702  000010          11$:    MOV     #8.,R2          ;8 CLOCKS
4868  024400  052710  000002          12$:    BIS     #MCLK,@R0       ;SET CLOCK
4869  024404  042710  000002                  BIC     #MCLK,@R0       ;CLEAR CLOCKS
4870  024410  005302                          DEC     R2              ;COUND DOWN
4871  024412  001372                          BNE     12$             ;BRANCH IF 8 NOT DONE
4872  024414  052710  000012                  BIS     #MSTCK!MCLK,@R0 ;SET SECTOR AND CLOCK
4873  024420  042710  000012                  BIC     #MSTCK!MCLK,@R0 ;CLEAR SECTOR AND CLOCK
4874
4875                                          ;*NOW ALL REGISTERS WILL BE COMPARED
4876                                          ;*SO FILL EXPECTED VALUE INTO SAVED LOCATIONS
4877
4878  024424  052737  100000  003162          BIS     #SC,@#REINTO+6  ;INCLUDE SC IN SAVED RHCS1
4879  024432  053737  002016  003200          BIS     @#ATTENT,@#REINTO+24 ;FILL APPROPIATE ATTENTION
4880                                                                  ;IN SAVED RHAS
4881  024440  052737  000001  003202          BIS     #DMD,@#REINTO+26 ;SET DMD IN RHMR SAVED
4882  024446  052737  100000  003204          BIS     #ATA,@#REINTO+30 ;SET ATA IN RHDS1 SAVED
4883  024454  013737  001200  003216          MOV     @#$TMP1,@#REINTO+42 ;MOVE EXPECTED VALUE
4884                                                                  ;INTO RHLA SAVED
4885
4886                                          ;*AFTER SEARCH COMMAND
4887                                          ;*SAVE REGISTERS AGAIN SO THAT COMPARES CAN BE DONE
4888
4889  024462  004037  043226                  JSR     R0,@#SAVER      ;SAVE
4890  024466  001632                          RHWC                    ;FROM
4891  024470  002110                          WRFROM                  ;TO
4892  024472  000023                          19.                     ;NUMBER
4893
4894                                          ;*AS UPPER BYTE OF RHAS CAN BE CHANGING IN A DUAL PORT
4895                                          ;*OPERATION THE UPPER BYTE OF RHAS WILL BE SAVED AS IS
4896                                          ;*SO THAT THE COMPARES ARE ONLY VALID FOR THE LOWER BYTE
4897  024474  113737  003201  002135          MOVB    @#REINTO+25,@#WRFROM+25;SAVE UPPER RHAS
4898
4899
4900                                          ;*COMPARE REGISTERS BEFORE SEARCH WITH AFTER
4901
4902  024502  004037  043430                  JSR     R0,@#COMPAR     ;COMPAR
4903  024506  003154                          REINTO                  ;GO BUFFER
4904
4905  024510  002110                          WRFROM                  ;TEST BUFFER
4906  024512  000022                          18.                     ;NUMBER
4907  024514  024522                          13$                     ;RETURN FOR ERROR
4908  024516  024522                          13$                     ;SAME
4909  024520  024542                          14$                     ;RETURN FOR GOOD COMPARISON
4910  024522  013705  047320          13$:    MOV     @#ERWORD,R5     ;GETTING READY TO INDEX
```

```
 4911  024526  060505                          ADD     R5,R5              ;DOUBLE ERROR WORD
 4912  024530  016537  001630  042204          MOV     RHWC-2(R5),@#REGADR ;FAILING REG. ADDRESS
 4913  024536  104001                          ERROR   1                  ;CONTENTS OF REGISTER
 4914  024540  000207                          RTS     PC                 ;CHANGED AT END OF
 4915  024542                        14$:                                 ;SEARCH
 4916
```

```
4917
4918                                    ;;*****************************************************************
4919                                    ;*TEST 55        MAKE CURRENT CYLINDER = 0
4920                                    ;;*****************************************************************
4921  024542  000004                    TST55:  SCOPE
4922  024544  012706  001000                    MOV     #STACK,SP       ;RESET STACK
4923  024550  012737  000055  002032            MOV     #TTNO,@#TSTNM   ;THIS SAVES TEST NUMBER
4924  024556  004737  042534                    JSR     PC,@#CLDISK     ;INIT DRIVE
4925  024562  012777  000001  155070            MOV     #DMD,@RHMR      ;SET DIAGNOSTIC MODE
4926  024570  004037  045106                    JSR     R0,@#MAKECYL    ;SUBROUTINE TO GIVE A SEEK
4927                                                                    ;COMMAND FOLOWED BY AN INIT
4928                                                                    ;THIS SHOULD CHANGE RHCC
4929  024574  000000                            0                       ;CHANGE RHCC TO 0
4930
```

ED

```
4931
4932
4933
4934                                    .SBTTL  READ/WRITE ADDRESSING VIA RHMP
4935
4936
4937                                    ;;*******************************************************************
4938                                    ;*TEST 56        WRITE HEADER AND DATA 1
4939
4940                                    ;**     WRITE CYLINDER 0, FORMAT 16 BIT PER WORD
4941                                    ;**     TRACK 0, SECTOR 0, KEYS 0, NUMBER OF WORDS 256 OF 0'S
4942                                    ;**     AS EVERYTHING IS ZERO THIS PROVES VERY LITTLE
4943                                    ;**     ANY ERROR LOGIC INDICATIONS ARE NOT CONCLUSIVE ON FIRST PASS
4944                                    ;**     BECAUSE ERROR LOGIC HAS NOT BEEN CHECKED YET (ON FIRST PASS)
4945
4946                                    ;;*******************************************************************
4947  024576  000004                    TST56:  SCOPE
4948  024600  012706  001000                    MOV     #STACK,SP       ;RESET STACK
4949  024604  012737  000056  002032            MOV     #TTNO,@#TSTNM   ;THIS SAVES TEST NUMBER
4950
4951  024612  012700  051020                    MOV     #SECGAP,R0      ;POINTER
4952  024616  012701  000460                    MOV     #304.,R1        ;COUNTER
4953  024622  012720  177777            1$:     MOV     #-1,(R0)+       ;CLEAR "DISK" AREA TO ALL ONES
4954  024626  005301                            DEC     R1              ;
4955  024630  001374                            BNE     1$              ;
4956  024632  004737  042534                    JSR     PC,CLDISK       ;THIS IS USED TO SET UP GENERAL
4957                                                                    ;REGISTER CORRESPONDENCE
4958
4959                                    ;*THESE ARE TO SET UP FOR DISKLESS USE ONLY
4960
4961  024636  012737  010000  052336            MOV     #FMT22,@#WCYL   ;FORMAT 22=16 BIT WORDS AND
4962                                                                    ;CYLINDER 0
4963  024644  005037  052340                    CLR     @#WSECTR        ;TRACK=0, SECTOR=0
4964  024650  005037  052342                    CLR     @#WKEY1         ;KEY1=0
4965  024654  005037  052344                    CLR     @#WKEY2         ;KEY2=0
4966  024660  012737  000400  052376            MOV     #256.,@#FWWORD  ;256 DATAWORDS
4967  024666  004537  043742                    JSR     R5,@#CRC        ;GO TO CALCULATE CRC
4968  024672  052336                            WCYL
4969  024674  052346                            GCRC
4970
4971                                    ;*THESE ARE REGULAR SETUPS FOR RH11 & "WRFROM" OUTPUT BUFFER
4972
4973  024676  012777  177374  154726            MOV     #-260.,@RHWC    ;256 DATA WORDS 4 HEADER WORDS
4974  024704  012700  002110                    MOV     #WRFROM,R0      ;BUS ADDRESS TO BE
4975  024710  010077  154720                    MOV     R0,@RHBA        ;BUFFER "WRFROM"
4976  024714  012705  000403                    MOV     #259.,R5        ;COUNTER
4977  024720  012720  010000                    MOV     #FMT22,(R0)+    ;FORMAT =16 BIT WORD
4978
4979  024724  005020                    2$:     CLR     (R0)+           ;SECTOR=0, TRACK=0,KEYS=0, ALL DATA=0
4980  024726  005305                            DEC     R5              ;& CYLINDER=0....SO CLEAR ALL "WRFROM"
4981  024730  001375                            BNE     2$              ;CONTINUE IF ALL 259 NOT COMPLETE
4982  024732  005077  154706                    CLR     @RHDST          ;TRACK=0, SECTOR=0
4983
4984
4985  024736  004737  042570                    JSR     PC,@#CHECKT     ;CHECK THAT DVA,RDY,DPR,DRY = 1
4986  024742  104401  062145                    TYPE    ,CPHALT         ;AND THAT NO OTHERS = 1.  CANNOT CON-
```

F10

```
4987
4988  024746  000000                          HALT                    ;STOP THE TEST
4989
4990  024750  013711  002064                  MOV     @#WRIFOR,@R1    ;GET READY FOR WRITE HEADER
4991                                                                    ;AND DATA WITH 62 IN RHCS1
4992  024754  005037  002006                  CLR     @#ERFLG$        ;CLEAR ERROR FLAG
4993  024760  012777  010000  154662          MOV     #FMT22,@RHOF    ;FORMAT BIT=1 16 BIT WORDS
4994  024766  005077  154660                  CLR     @RHCA           ;CYLINDER 0
4995  024772  004737  052162                  JSR     PC,@#COMWHD     ;WRITE HEADER AND DATA FROM "WRFROM"
4996                                                                    ;INTO THE RHMR REGISTER AND BACK INTO
4997                                                                    ;CORE "DISK" AREA
4998
4999                                          ;*IF THE PROGRAM COMES BACK HERE WITHOUT ERROR
5000                                          ;*PRINT OUTS FROM THE "COMWHD" ROUTINE THAT MEANS
5001                                          ;*ALL HEADER ON DISK IS GOOD IE. ONLY DATA IS
5002                                          ;*TO BE CHECKED TO SEE IF IT IS ZERO
5003
5004                                          ;*AND WRITE DATA GAP AND TOLERANCE GAP TO SEE IF THEY
5005                                          ;*ARE ALL ZEROS (ECC1 AND ECC2 MAY NOT BE 0) AS WELL AS
5006                                          ;*CHECKING RHWC FOR ZERO
5007
5008  024776  017737  154630  001126          MOV     @RHWC,$BDDAT    ;MOVE WORD COUNTER INTO BAD DATA
5009  025004  001401                          BEQ     5$              ;SHOULD HAVE COUNTED UP TO ZERO
5010  025006  104040                          ERROR   40              ;RHWC DID NOT = 0 AFTER A WRITE
5011                                                                    ;HEADER AND DATA
5012
5013  025010  005737  002006          5$:     TST     @#ERFLG$        ;HAVE ANY ERRORS OCCURED?
5014  025014  001034                          BNE     TST57           ;;BRANCH IF YES
5015  025016  004737  042760                  JSR     PC,@#CHECKE     ;CHECK THAT BITS = 1
5016  025022  104401  062145                  TYPE    ,CPHALT         ;CANNOT CONTINUE TESTING IF THEY DON'T
5017  025026  000000                          HALT                    ;STOP THE TEST
5018  025030  005037  052116                  CLR     @#WECC1         ;CLEAR ECC
5019  025034  005037  052120                  CLR     @#WECC2
5020
5021                                          ;*REINTO BUFFER IS FILLED WITH EXPECTED DATA OF ALL 0'S
5022
5023  025040  004037  042452                  JSR     R0,@#CLAREA     ;CLEAR "REINTO"
5024  025044  003154                                  REINTO          ;FROM
5025  025046  004214                                  REINTO+<272.*2> ;TO
5026  025050  000000                          .WORD   0               ;FILL WITH ZEROS
5027  025052  005037  002006                  CLR     @#ERFLG$        ;CLEAR ERROR FLAG
5028
5029
5030                                          ;*COMPARE "REINTO" BUFFER WITH "DISK" BUFFER
5031  025056  004037  043430                  JSR     R0,@#COMPAR     ;CHECK
5032  025062  003154                          REINTO                  ;GOOD BUFFER
5033  025064  051116                          DISK                    ;TEST BUFFER
5034  025066  000421                          273.                    ;NUMBER OF WORDS CHECKED
5035  025070  025076                          3$                      ;RETURN POINT FOR ERROR HEADER
5036  025072  025102                          4$                      ;RETURN POINT FOR ERROR DATA
5037  025074  025106                          TST57                   ;RETURN FOR GOOD COMPARISON
5038  025076  104007          3$:     ERROR   7               ;READ ERROR10 NEXT
5039  025100  000207                  RTS     PC              ;RETURN TO "COMPAR"
5040  025102  104010          4$:     ERROR   10              ;WORD NOS 1 TO 256 ARE
5041                                                          ;DATA WORDS
5042                                                          ;257 AND 258 ARE ECC
```

G10

```
 5043                                                          ;ZEROED OUT
 5044                                                          ;259 TO 273 TOLERANCE GAP
 5045  025104  000207                        RTS     PC
 5046
```

```
5047
5048                                    ;;*****************************************************************
5049                                    ;*TEST 57        WRITE HEADER AND DATA 2
5050
5051                                    ;**     WRITE CYLINDER0, FORMAT 16 BITS PER WORD
5052                                    ;**     TRACK 0, SECTOR 1, KEYS 0, NUMBER OF WORDS 256
5053                                    ;**     OF ALL ONES.
5054                                    ;**     ANY ERROR LOGIC INDICATIONS ARE NOT CONCLUSIVE ON FIRST PASS
5055                                    ;**     BECAUSE ERROR LOGIC HAS NOT BEEN CHECKED YET (ON FIRST PASS)
5056
5057                                    ;;*****************************************************************
5058  025106  000004                    TST57:  SCOPE
5059  025110  012706  001000                    MOV     #STACK,SP       ;RESET STACK
5060  025114  012737  000057  002032            MOV     #TTNO,@#TSTNM   ;THIS SAVES TEST NUMBER
5061
5062  025122  012700  051020                    MOV     #SECGAP,R0      ;POINTER
5063  025126  012701  000460                    MOV     #304.,R1        ;COUNTER
5064  025132  005020                    1$:     CLR     (R0)+           ;CLEAR SIMULATED "DISK" AREA IN CORE
5065  025134  005301                            DEC     R1              ;
5066  025136  001375                            BNE     1$              ;
5067  025140  004737  042534                    JSR     PC,@#CLDISK     ;THIS IS USED TO SET GENERAL REGISTERS
5068
5069                                            ;*THESE ARE TO BE SETUP FOR DISKLESS USE ONLY
5070
5071  025144  012737  010000  052336            MOV     #FMT22,@#WCYL   ;FORMAT 22 = 16 BIT WORDS AND
5072                                                                    ;CYLINDER 0
5073  025152  012737  000001  052340            MOV     #1,@#WSECTR     ;TRACK=0, SECTOR=1
5074  025160  005037  052342                    CLR     @#WKEY1         ;KEY1=0
5075  025164  005037  052344                    CLR     @#WKEY2         ;KEY2=0
5076  025170  012737  000400  052376            MOV     #256.,@#FNWORD  ;256 DATA WORDS
5077  025176  004537  043742                    JSR     R5,@#CRC        ;GO TO CALCULATE CRC
5078  025202  052336                            WCYL
5079  025204  052346                            GCRC
5080
5081                                            ;*THESE ARE REGULAR SETUPS FOR THE RH11 AND "WRFROM" BUFFER
5082
5083  025206  012777  177374  154416            MOV     #-260.,@RHWC    ;256 DATA WORDS 4 HEADER WORDS
5084  025214  012700  002110                    MOV     #WRFROM,R0      ;THESE TWO INSTRUCTIONS GETS
5085  025220  010077  154410                    MOV     R0,@RHBA        ;ADDR. OF WRFROM INTO R0 AND
5086                                                                    ;BUS ADDRESS REGISTER
5087  025224  012720  010000                    MOV     #FMT22,(R0)+    ;FORMAT=16 BIT WORDS
5088                                                                    ;CYLINDER=0
5089  025230  012720  000001            2$:     MOV     #1,(R0)+        ;TRACK=0, SECTOR=1, KEYS=0
5090  025234  005020                            CLR     (R0)+           ;KEY1=0
5091  025236  005020                            CLR     (R0)+           ;KEY2=0
5092  025240  012705  000400                    MOV     #256.,R5        ;COUNTER
5093  025244  012720  177777            3$:     MOV     #-1,(R0)+       ;MOVE ALL ONES FOR DATA
5094  025250  005305                            DEC     R5
5095  025252  001374                            BNE     3$              ;BRANCH IF DATA NOT COMPLETE
5096  025254  012777  000001  154362            MOV     #1,@RHDST       ;TRACK=0 SECTOR=1
5097
5098  025262  004737  042570                    JSR     PC,@#CHECKT     ;CHECK THAT DVA,RDY,DPR,DRY = 1
5099  025266  104401  062145                    TYPE    ,CPHALT         ;AND THAT NO OTHERS = 1.  CANNOT CON-
5100                                                                    ;TINUE TESTING IF BOTH AREN'T TRUE
5101  025272  000000                            HALT                    ;STOP THE TEST
5102
```

I16

```
5103  025274  013711  002064                  MOV     @#WRIFOR,@R1    ;GET READY FOR WRITE HEADER AND
5104                                                                    ;DATA WITH 62 IN RHCS1
5105  025300  005037  002006                  CLR     @#ERFLG$        ;CLEAR ERROR FLAG
5106  025304  012777  010000  154336          MOV     #FMT22,@RHOF    ;FORMAT BIT=1 (16 BIT WORDS)
5107  025312  005077  154334                  CLR     @RHCA           ;CYLINDER =0
5108  025316  004737  052162                  JSR     PC,@#COMWHD     ;WRITE HEADER AND DATA INTO "DISK" AREA
5109
5110                                          ;*IF THE PROGRAM COMES BACK HERE WITHOUT ERROR PRINTOUTS
5111                                          ;*FROM THE "COMWHD" ROUTINE THAT MEANS ALL THE HEADER ON "DISK"
5112                                          ;*IS GOOD IE. ONLY THE DATA IS TO BE CHECKED TO SEE IF IT IS
5113                                          ;*ALL ONES AND WRITE DATA GAP AND TOLERANCE GAP TO SEE IF
5114                                          ;*THEY ARE ALL ZEROS, - ECC1 AND ECC2 ARE NOT CHECKED.
5115
5116                                          ;*RHWC IS CHECKED TO BE = 0
5117
5118  025322  017737  154304  001126          MOV     @RHWC,$BDDAT    ;LOAD WORD COUNTER JUST IN CASE
5119  025330  001401                          BEQ     6$              ;SHOULD BE = 0
5120  025332  104040                          ERROR   40              ;RHWC DOES NOT = 0 AFTER A WRITE
5121                                                                  ;HEADER AND DATA IS COMPLETED
5122
5123  025334  005737  002006          6$:     TST     @#ERFLG$        ;HAVE ANY ERRORS OCCURRED?
5124  025340  001041                          BNE     TST60           ;;BRANCH IF YES
5125  025342  004737  042760                  JSR     PC,@#CHECKE     ;CHECK THAT BITS = 1
5126  025346  104401  062145                  TYPE    ,CPHALT         ;CANNOT CONTINUE TESTING IF THEY DON'T
5127  025352  000000                          HALT                    ;STOP THE TEST
5128  025354  005037  052116                  CLR     @#WECC1         ;CLEAR ECC
5129  025360  005037  052120                  CLR     @#WECC2         ;CLEAR ECC
5130
5131                                          ;*FILL "REINTO" BUFFER WITH EXPECTED DATA OF ALL 1'S
5132
5133  025364  004037  042452                  JSR     R0,@#CLAREA     ;FILL REINTO BUFFER
5134  025370  003154                                  REINTO          ;FROM
5135  025372  004152                                  REINTO+<255.*2> ;TO
5136  025374  177777                          .WORD   -1              ;DATA
5137  025376  004037  042452                  JSR     R0,@#CLAREA     ;FILL REST
5138  025402  004154                          REINTO+<256.*2>         ;FROM
5139  025404  004214                          REINTO+<272.*2>         ;TO
5140  025406  000000                          0                       ;DATA
5141
5142  025410  005037  002006                  CLR     @#ERFLG$        ;CLEAR ERROR FLAG
5143
5144                                          ;*NOW COMPARE "DISK" BUFFER WITH "REINTO" BUFFER IN CORE
5145
5146  025414  004037  043430                  JSR     R0,@#COMPAR     ;CHECK
5147  025420  003154                          REINTO                  ;GOOD BUFFER
5148  025422  051116                          DISK                    ;TEST BUFFER
5149  025424  000421                          273.                    ;NUMBER OF WORDS CHECKED
5150  025426  025434                          4$                      ;RETURN POINT FOR ERROR HEADER
5151  025430  025440                          5$                      ;RETURN POINT FOR ERROR DATA
5152  025432  025444                          TST60                   ;RETURN FOR GOOD COMPARISON
5153  025434  104007                  4$:     ERROR   7               ;READ ERROR 10 NEXT
5154  025436  000207                          RTS     PC              ;RETURN TO COMPARE
5155  025440  104010                  5$:     ERROR   10              ;WORD NOS 1 TO 256 ARE
5156                                                                  ;DATA WORDS
5157                                                                  ;WORD NOS 257 AND 258
5158                                                                  ;ARE ECC WHICH HAVE BEEN
```

J10

```
                                                        ;ZEROED
                                                        ;WORD NOS 259
                                                        ;IS DATA GAP
                                                        ;WORD NOS 260 TO 273
                                                        ;ARE TOLERANCE GAP
025442  000207                        RTS     PC      ;RETURN TO COMPARE
```

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

                                ;;*****************************************************************
025444  000004                  TST60:  SCOPE
025446  012706  001000                  MOV     #STACK,SP       ;RESET STACK
025452  012737  000060  002032          MOV     #TTNO,@#TSTNM   ;THIS SAVES TEST NUMBER

025460  012700  051020                  MOV     #SECGAP,R0      ;POINTER
025464  012701  000460                  MOV     #304.,R1        ;COUNTER
025470  012720  000377          1$:     MOV     #377,(R0)+      ;LOAD SIMULATED "DISK" AREA WITH 377
025474  005301                  DEC     R1                      ;
025476  001374                  BNE     1$                      ;
025500  004737  042534                  JSR     PC,CLDISK       ;THIS IS USED TO SET UP GENERAL
                                                                ;REGISTER CORRESPONDENCE

                                ;*THESE ARE TO BE SETUP FOR DISKLESS USE ONLY

025504  012737  010000  052336          MOV     #FMT22,@#WCYL   ;FORMAT 22=16 BIT WORDS AND
                                                                ;CYLINDER 0
025512  012737  000401  052340          MOV     #401,@#WSECTR   ;TRACK=1, SECTOR=1
025520  005037  052342                  CLR     @#WKEY1         ;KEY1=0
025524  005037  052344                  CLR     @#WKEY2         ;KEY2=0
025530  012737  000400  052376          MOV     #256.,@#FWWORD  ;256 DATA WORDS
025536  004537  043742                  JSR     R5,@#CRC        ;GO TO CALCULATE CRC
025542  052336                  WCYL
025544  052346                  GCRC

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

025546  012777  177374  154056          MOV     #-260.,@RHWC    ;256 DATA WORDS 4 HEADER WORDS
025554  012700  002110                  MOV     #WRFROM,R0      ;THESE TWO INSTRUCTIONS GET
                                                                ;ADDR. OF WRFROM INTO R0
025560  010077  154050                  MOV     R0,@RHBA        ;AND BUS ADDRESS REGISTER

025564  012720  010000                  MOV     #FMT22,(R0)+ .  ;FORMAT=16 BIT WORDS
                                                                ;CYLINDER=0
025570  012720  000401          2$:     MOV     #401,(R0)+      ;TRACK=1, SECTOR=1, KEYS=0
025574  005020                  CLR     (R0)+           ;KEY1=0
025576  005020                  CLR     (R0)+           ;KEY2=0
025600  012705  000400                  MOV     #256.,R5        ;COUNTER
025604  012720  052525          3$:     MOV     #052525,(R0)+   ;MOVE ALTERNATE ONES FOR DATA
025610  005305                  DEC     R5              ;COUNT
025612  001374                  BNE     3$              ;BRANCH IF DATA NOT COMPLETE
025614  012777  000401  154022          MOV     #401,@RHDST     ;TRACK=1 SECTOR=1

025622  004737  042570                  JSR     PC,@#CHECKT     ;CHECK THAT DVA,RDY,DPR,DRY = 1
025626  104401  062145                  TYPE    ,CPHALT         ;AND THAT NO OTHERS = 1.  CANNOT CON-
                                                                ;TINUE TESTING IF BOTH AREN'T TRUE
025632  000000                  HALT                            ;STOP THE TEST
```

L10

```
5224
5225  025634  013711  002064                  MOV     @#WRIFOR,@R1    ;GET READY FOR WRITE HEADER
5226                                                                    ;AND DATA WITH 62 IN RHCS1
5227  025640  005037  002006                  CLR     @#ERFLG$        ;CLEAR ERROR FLAG
5228  025644  012777  010000  153776          MOV     #FMT22,@RHOF    ;FORMAT BIT=1(16 BIT WORDS
5229  025652  005077  153774                  CLR     @RHCA           ;CYLINDER=0
5230  025656  004737  052162                  JSR     PC,@#COMWHD     ;WRITE HEADER AND DATA INTO "DISK" CORE
5231
5232                                          ;*IF THE PROGRAM COMES BACK HERE WITHOUT ERROR PRINTOUTS
5233                                          ;*FROM THE "COMWHD" ROUTINE THAT MEANS ALL HEADER IN
5234                                          ;*"DISK" IS GOOD.  DATA IS TO BE CHECKED TO SEE
5235                                          ;*IF IT IS ALL 052525 AND WRITE DATA GAP AND
5236                                          ;*TOLERANCE GAP TO SEE IF THEY ARE ALL ZEROS.
5237
5238                                          ;*RHWC IS CHECKED TO BE = 0
5239
5240  025662  017737  153744  001126          MOV     @RHWC,$BDDAT    ;LOAD AND TEST FOR ZERO
5241  025670  001401                          BEQ     6$              ;RHWC SHOULD = 0
5242  025672  104040                          ERROR   40              ;RHWC DID NOT = 0 AFTER A WRITE
5243                                                                  ;HEADER AND DATA WAS COMPLETED
5244
5245                                          ;*ONLY ECC1 AND ECC2 ARE NOT CHECKED
5246
5247  025674  005737  002006          6$:     TST     @#ERFLG$        ;HAVE ANY ERRORS OCCURED?
5248  025700  001041                          BNE     TST61           ;;BRANCH IF YES
5249  025702  004737  042760                  JSR     PC,@#CHECKE     ;CHECK THAT BITS = 1
5250  025706  104401  062145                  TYPE    ,CPHALT         ;CANNOT CONTINUE TESTING IF THEY DON'T
5251  025712  000000                          HALT                    ;STOP THE TEST
5252  025714  005037  052116                  CLR     @#WECC1         ;CLEAR ECC
5253  025720  005037  052120                  CLR     @#WECC2         ;CLEAR ECC
5254
5255                                          ;*FILL "REINTO" BUFFER WITH EXPECTED DATA
5256
5257  025724  004037  042452                  JSR     R0,@#CLAREA     ;FILL REINTO BUFFER
5259  025730  003154                          REINTO                  ;FROM
5259  025732  004152                          REINTO+<255.*2>         ;TO
5260  025734  052525                          .WORD   52525           ;DATA
5261  025736  004037  042452                  JSR     R0,@#CLAREA     ;FILL REST
5262  025742  004154                                  REINTO+<256.*2> ;FROM
5263  025744  004214                          REINTO+<272.*2>         ;TO
5264  025746  000000                          .WORD   0               ;DATA
5265  025750  005037  002006                  CLR     @#ERFLG$        ;CLEAR ERROR FLAG
5266
5267                                          ;*NOW COMPARE "DISK" BUFFER WITH "REINTO" BUFFER IN CORE
5268
5269  025754  004037  043430                  JSR     R0,@#COMPAR     ;CHECK
5270  025760  003154                          REINTO                  ;GOOD BUFFER
5271  025762  051116                          DISK                    ;TEST BUFFER
5272  025764  000421                          273.                    ;NUMBER OF WORDS CHECKED
5273  025766  025774                          4$                      ;RETURN POINT FOR ERROR HEADER
5274  025770  026000                          5$                      ;RETURN POINT FOR ERROR DATA
5275  025772  026004                          TST61                   ;RETURN FOR GOOD COMPARISON
5276  025774  104007                  4$:     ERROR   7               ;READ ERROR 10 NEXT
5277  025776  000207                          RTS     PC              ;RETURN TO COMPARE
5278  026000  104010                  5$:     ERROR   10              ;WORD NOS 1 TO 256 ARE
5279                                                                  ;DATA WORDS
```

M10

```
5280                                                    ;WORD NOS 257 AND 258
5281                                                    ;ARE ECC WHICH HAVE BEEN
5282                                                    ;ZEROED
5283                                                    ;WORD NOS 259
5284                                                    ;IS DATA GAP
5285                                                    ;WORD NOS 260 TO 273
5296                                                    ;ARE TOLERANCE GAP
5287  026002  000207                    RTS     PC      ;RETURN TO COMPARE
5288
5289
5290
5291
```

```
5292
5293                                ;;*******************************************************************
5294                                ;*TEST 61       PROGRAM ERROR RHCS2 #10
5295
5296                                ;**     WRITE CYLINDER 0, FORMAT 16 BIT PER WORD
5297                                ;**     TRACK 0, SECTOR 0, KEYS 0, NUMBER OF WORDS 256 OF 0'S
5298                                ;**     WHILE GO BIT IS SET ANOTHER GO IS GIVEN THIS SHOULD SET
5299                                ;**     PROGRAM ERROR
5300
5301                                ;;*******************************************************************
5302  026004  000004                TST61:  SCOPE
5303  026006  012706  001000               MOV     #STACK,SP       ;RESET STACK
5304  026012  012737  000061  002032       MOV     #TTNO,@#TSTNM   ;THIS SAVES TEST NUMBER
5305
5306  026020  012700  051020               MOV     #SECGAP,R0      ;POINTER
5307  026024  012701  000460               MOV     #304.,R1        ;COUNTER
5308  026030  012720  177777        1$:    MOV     #-1,(R0)+       ;CLEAR DISK AREA TO ALL ONES.
5309  026034  005301                       DEC     R1              ;
5310  026036  001374                       BNE     1$              ;
5311  026040  004737  042534               JSR     PC,CLDISK       ;THIS IS USED TO SET GENERAL
5312                                                                ;REGISTERS
5313
5314                                       ;*THESE ARE TO SET UP FOR DISKLESS USE ONLY
5315
5316  026044  012737  010000  052336       MOV     #FMT22,@#WCYL   ;FORMAT 22=16 BITWORDS AND
5317                                                                ;CYLINDER 0
5318  026052  005037  052340               CLR     @#WSECTR        ;TRACK=0, SECTOR=0
5319  026056  005037  052342               CLR     @#WKEY1         ;KEY1=0
5320  026062  005037  052344               CLR     @#WKEY2         ;KEY2=0
5321  026066  012737  000400  052376       MOV     #256.,@#FNWORD  ;256 DATAWORDS
5322  026074  004537  043742               JSR     R5,@#CRC        ;GO TO CALCULATE CRC
5323  026100  052336                       WCYL
5324  026102  052346                       GCRC
5325
5326                                       ;*THESE ARE REGULAR SETUPS
5327
5328  026104  012777  177374  153520       MOV     #-260.,@RHWC    ;256 DATA WORDS 4 HEADER WORDS
5329  026112  012700  002110               MOV     #WRFROM,R0      ;FROM BUFFER "WRFROM"
5330  026116  010077  153512               MOV     R0,@RHBA        ;IN BUS ADDRESS
5331  026122  012705  000403               MOV     #259.,R5        ;COUNTER
5332  026126  012720  010000               MOV     #FMT22,(R0)+    ;FORMAT =16 BIT WORD
5333                                                                ;CYLINDER=0
5334  026132  005020                2$:    CLR     (R0)+           ;SECTOR=0, TRACK=0,KEYS=0, ALL DATA=0
5335  026134  005305                       DEC     R5              ;COUNT
5336  026136  001375                       BNE     2$              ;BRANCH IF ALL 259 NOT COMPLETE
5337  026140  005077  153500               CLR     @RHDST          ;TRACK=0, SECTOR=0
5338
5339  026144  004737  042570               JSR     PC,@#CHECKT     ;CHECK THAT DVA,RDY,DPR,DRY = 1
5340  026150  104401  062145               TYPE    ,CPHALT         ;AND THAT NO OTHERS = 1.  CANNOT CON-
5341                                                                ;TINUE TESTING IF BOTH AREN'T TRUE
5342  026154  000000                       HALT                    ;STOP THE TEST
5343
5344  026156  013711  002064               MOV     @#WRIFOR,@R1    ;GET READY FOR WRITE HEADER
5345                                                                ;AND DATA WITH 62 IN RHCS1
5346  026162  005037  002006               CLR     @#ERFLG$        ;CLEAR ERROR FLAG
5347  026166  012777  010000  153454       MOV     #FMT22,@RHOF    ;FORMAT BIT=1 16 BIT WORDS
```

```
 5348  026174  005077  153452                  CLR     @RHCA           ;CYLINDER 0
 5349  026200  012777  000001  153452          MOV     #DMD,@RHMR      ;SET DIAGNOSTIC MODE
 5350  026206  052777  000001  153424          BIS     #GO,@RHCS1      ;GO
 5351  026214  000240                          NOP
 5352  026216  052777  000001  153414          BIS     #GO,@RHCS1      ;THIS GO SHOULD SET PGE
 5353
 5354  026224  004737  042140                  JSR     PC,@#PUTREG     ;SAVE REGISTERS
 5355  026230  032737  002000  001714          BIT     #PGE,@#CS1      ;IS PGE SET
 5356  026236  001404                          BEQ     3$              ;BRANCH IF GOOD
 5357  026240  013737  001636  001122          MOV     @#RHCS2,@#$BDADR
 5358  026246  104037                          ERROR   37              ;PGE DID NOT SET WHEN A WRITE
 5359                                                                  ;WAS ATTEMPTED WITH ONE IN PROGRESS
 5360  026250                          3$:
```

```
5361
5362                    ;;*****************************************************************
5363                    ;**     THESE TESTS ARE THROUGH THE MAINTAINABILITY REGISTER - RHMR
5364
5365                    ;**     THE SECTOR GAP AND SYNC BYTE ARE ALWAYS READ AS
5366                    ;**     ZEROS AND 144000 NO MATTER WHAT IS IN THE SIMULATED DISK AREA
5367                    ;**     TAGGED SECGAP: AND WSSYNC:
5368
5369                    ;**     THE HEADER CONSISTING OF CYLINDER ADDRESS, SECTOR,
5370                    ;**     TRACK AND THE KEYS ARE READ FROM LOCATION
5371                    ;**     CYL:, SECTOR:, KEY1:, AND KEY2 AND NOT FROM
5372                    ;**     HEADER: ON SIMULATED DISK
5373
5374                    ;**     CRC IS READ FROM SIMULATED DISK LOCATION WCRC:
5375                    ;**     HEADER GAP IS ALWAYS READ AS ZEROS NO MATTER
5376                    ;**     WHAT IS ON THE SIMULATED DISK AREA
5377
5378                    ;**     THE DATA SYNC IS READ  FROM HDWSYN:
5379                    ;**     ON SIMULATED DISK
5380
5381                    ;**     ALL DATA IS READ FROM SIMULATED DISK DISK:
5382                    ;;*****************************************************************
```

```
5383
5384
5385
5386
5387                                    ;;*******************************************************************
5388                                    ;*TEST 62       READ HEADER AND DATA 1
5389                                    ;**      READ CYLINDER 0 FORMAT 16 BITS PER WORD
5390                                    ;**      TRACK 0, SECTOR 0, KEYS 0, 256 WORDS OF 0
5391                                    ;**      ANY ERROR LOGIC INDICATIONS ARE NOT CONCLUSIVE ON FIRST PASS
5392                                    ;**      BECAUSE ERROR LOGIC HAS NOT BEEN CHECKED YET (ON FIRST PASS)
5393
5394                                    ;;*******************************************************************
5395  026250  000004                    TST62:  SCOPE
5396  026252  012706  001000                    MOV     #STACK,SP       ;RESET STACK
5397  026256  012737  000062  002032            MOV     #TTWO,@#TSTNM   ;THIS SAVES TEST NUMBER
5398
5399
5400                                    ;*      SETUP FOR WHAT IS TO BE READ
5401                                    ;*      HEADER CRC IS RESTORED FROM A SUBROUTINE
5402
5403  026264  012746  000000                    MOV     #0,-(SP)        ;DATA TO BE READ
5404  026270  012705  000400                    MOV     #256.,R5        ;COUNTER
5405  026274  012700  051116                    MOV     #DISK,R0        ;START OF SIMULATED DISK DATA
5406  026300  011620                    1$:     MOV     (SP),(R0)+      ;MOVE IN DATA ON TO SIMULATED DISK
5407  026302  005305                            DEC     R5              ;COUNT
5408  026304  001375                            BNE     1$              ;BRANCH IF 256 NOT COMPLETE
5409  026306  005726                            TST     (SP)+           ;UNDO -(SP)
5410  026310  012705  000021                    MOV     #17.,R5         ;2 ECC WORDS
5411                                                                    ;1 DATA GAP
5412                                                                    ;14 TOLERANCE GAP
5413  026314  005020                    2$:     CLR     (R0)+           ;CLEAR ECC, DATA GAP, AND
5414  026316  005305                            DEC     R5              ;TOLERANCE GAP
5415  026320  001375                            BNE     2$              ;BRANCH IF NOT COMPLETE
5416
5417
5418                                            ;*THESE ARE TO SETUP FOR DISKLESS USE ONLY
5419
5420  026322  012737  010000  047200            MOV     #01!FMT22,@#CYL ;16 BITS PER WORD
5421                                                                      ;CYLINDER 0, FORMAT 16 BITS
5422  026330  112737  000000  047203            MOVB    #0,@#SECOTR+1   ;TRACK 0
5423  026336  112737  000000  047202            MOVB    #0,@#SECOTR     ;SECTOR 0
5424  026344  012737  000000  047204            MOV     #0,@#KEY1       ;KEY1=0
5425  026352  012737  000000  047206            MOV     #0,@#KEY2       ;KEY2=0
5426  026360  012737  000400  047260            MOV     #256.,@#DAWORD  ;NO. OF DATA WORDS
5427  026366  005037  047210                    CLR     @#X             ;THIS IS A READ COMMAND
5428  026372  004537  043742                    JSR     R5,@#CRC        ;GO TO CALCULATE CRC
5429  026376  047200                            CYL
5430  026400  051100                            WCRC
5431
5432                                            ;*THESE ARE REGULAR SETUPS FOR RH11 AND "REINTO"
5433
5434  026402  004737  042534                    JSR     PC,@#CLDISK     ;SETUP GENERAL REGISTERS
5435  026406  012777  177374  153216            MOV     #-256.-4.,@RHWC ;256. DATA 4 HEADER WORDS
5436  026414  012777  003154  153212            MOV     #REINTO,@RHBA   ;STARTING ADDRESS OF READ BUFFER
5437  026422  112746  000000                    MOVB    #0,-(SP)        ;IN LOWER BYTE GET SECTOR
5438  026426  112766  000000  000001            MOVB    #0,1(SP)        ;GET TRACK IN HIGHER BYTE
```

E11

```
5439  026434  012677  153204                  MOV     (SP)+,@RHDST    ;TRACK/SECTOR IN RHDST
5440  026440  012777  014000  153202          MOV     #FMT22!ECI,@RHOF ;16 BITS PER WORD
5441                                                                    ;ECC CORRECTION INHIBIT
5442                                                                    ;BECAUSE ECC IS NOT GOING
5443                                                                    ;TO BE CHECKED
5444  026446  005077  153200                  CLR     @RHCA           ;CYLINDER 0
5445
5446  026452  004737  042570                  JSR     PC,@#CHECKT     ;CHECK THAT DVA,RDY,DPR,DRY = 1
5447  026456  104401  062145                  TYPE    ,CPHALT         ;AND THAT NO OTHERS = 1.  CANNOT CON-
5448                                                                    ;TINUE TESTING IF BOTH AREN'T TRUE
5449  026462  000000                          HALT                    ;STOP THE TEST
5450
5451  026464  013711  002070                  MOV     @#REFOR,@R1     ;READ HEADER AND DATA=72
5452  026470  005037  002006                  CLR     @#ERFLG$        ;CLEAR ERROR FLAG
5453  026474  004737  047040                  JSR     PC,@#COMHD      ;READ HEADER AND DATA
5454
5455
5456                                          ;*IF THE PROGRAM COMES BACK HERE WITHOUT ERROR PRINTOUTS
5457                                          ;*FROM THE "COMHD" ROUTINE THAT MEANS SECTOR GAP,
5458                                          ;*FIRST SYNC, HEADER, HEADER CRC, HEADER GAP AND
5459                                          ;*SYNC BYTE HAVE GONE BY AND SYNCS WERE CORRECTLY
5460                                          ;*DETECTED.
5461
5462
5463                                          ;*RHWC IS CHECKED TO BE = 0 AFTER THE READ OPERATION
5464
5465  026500  017737  153126  001126          MOV     @RHWC,$BDDAT    ;LOAD AND TEST RHWC
5466  026506  001401                          BEQ     20$             ;SHOULD = 0
5467  026510  104040                          ERROR   40              ;RHWC DOES NOT = 0 AFTER A READ
5468                                                                    ;HEADER AND DATA
5469
5470                                          ;*HEADER AND DATA ARE TO BE CHECKED.
5471                                          ;*IN CHECKING READ DATA THE WRITE FROM BUFFER
5472                                          ;*"WRFROM" IS FILLED WITH EXPECTED DATA AND
5473                                          ;*COMPARISONS ARE MADE.
5474
5475  026512  005737  002006          20$:    TST     @#ERFLG$        ;ANY ERRORS ALREADY THERE
5476  026516  001046                          BNE     TST63           ;BRANCH IF YES
5477  026520  004737  042760                  JSR     PC,@#CHECKE     ;CHECK THAT BITS = 1
5478  026524  104401  062145                  TYPE    ,CPHALT         ;CANNOT CONTINUE TESTING IF THEY DON'T
5479  026530  000000                          HALT                    ;STOP THE TEST
5480  026532  012700  002110                  MOV     #WRFROM,R0      ;GETTING READY TO FILL EXPECTED DATA
5481  026536  012720  010000                  MOV     #01!FMT22,(R0)+ ;CYLINDER 0
5482  026542  112746  000000                  MOVB    #0,-(SP)        ;IN LOWER BYTE GET SECTOR
5483  026546  112766  000000  000001          MOVB    #0,1(SP)        ;GET TRACK IN HIGHER BYTE
5484  026554  012620                          MOV     (SP)+,(R0)+     ;GET TRACK/SECTOR IN BUFFER
5485  026556  012720  000000                  MOV     #0,(R0)+        ;KEY1 IN BUFFER
5486  026562  012720  000000                  MOV     #0,(R0)+        ;KEY2 IN BUFFER
5487  026566  012701  000400                  MOV     #256.,R1        ;DATA WORD COUNTER
5488  026572  012702  000000                  MOV     #0,R2           ;DATA
5489  026576  010220                  3$:     MOV     R2,(R0)+        ;DATA INTO BUFFER
5490  026600  005301                          DEC     R1              ;COUNT
5491  026602  001375                          BNE     3$              ;BRANCH IF 256 NOT DONE
5492
5493                                          ;*NOW READ DATA BUFFER WILL BE CHECKED
5494
```

FN

```
 5495  026604  004037  043430                JSR     R0,@#COMPAR     ;CHECK
 5496  026610  002110                        WRFROM                  ;GOOD BUFFER
 5497  026612  003154                        REINTO                  ;TEST BUFFER
 5498  026614  000404                        4+256.                  ;NUMBER OF WORDS CHECKED
 5499  026616  026624                        4$                      ;RETURN POINT FOR ERROR HEADER
 5500  026620  026630                        5$                      ;RETURN POINT FOR ERROR DATA
 5501  026622  026634                        TST63                   ;RETURN FOR GOOD COMPARISON
 5502  026624  104004              4$:       ERROR   4               ;READ NEXT ERROR
 5503  026626  000207                        RTS     PC              ;RETURN TO "COMPAR"
 5504  026630  104005              5$:       ERROR   5               ;WORD NOS 1 TO 4 ARE
 5505                                                                ;HEADER WORDS
 5506                                                                ;5 TO 260 ARE DATA WORDS
 5507  026632  000207                        RTS     PC              ;RETURN TO "COMPAR"
 5508
 5509
 5510
 5511
 5512
```

```
5513
5514                                    ;;*****************************************************************
5515                                    ;*TEST 63       READ HEADER AND DATA 2
5516                                    ;**     READ CYLINDER 0 FORMAT 16 BITS PER WORD
5517                                    ;**     TRACK 0, SECTOR 1, KEYS 0, 256 WORDS OF 177777
5518                                    ;**     ANY ERROR LOGIC INDICATIONS ARE NOT CONCLUSIVE ON FIRST PASS
5519                                    ;**     BECAUSE ERROR LOGIC HAS NOT BEEN CHECKED YET (ON FIRST PASS)
5520
5521                                    ;;*****************************************************************
5522  026634  000004                    TST63:  SCOPE
5523  026636  012706  001000                    MOV     #STACK,SP       ;RESET STACK
5524  026642  012737  000063  002032            MOV     #TTNO,@#TSTNM   ;THIS SAVES TEST NUMBER
5525
5526
5527                                    ;*      SETUP FOR WHAT IS TO BE READ
5528                                    ;*      HEADER CRC IS RESTORED FROM A SUBROUTINE
5529
5530  026650  012746  177777                    MOV     #-1,-(SP)       ;DATA TO BE READ
5531  026654  012705  000400                    MOV     #256.,R5        ;COUNTER
5532  026660  012700  051116                    MOV     #DISK,R0        ;START OF SIMULATED DISK DATA
5533  026664  011620                    1$:     MOV     (SP),(R0)+      ;MOVE IN DATA ON TO SIMULATED DISK
5534  026666  005305                            DEC     R5              ;COUNT
5535  026670  001375                            BNE     1$              ;BRANCH IF 256 NOT COMPLETE
5536  026672  005726                            TST     (SP)+           ;UNDO -(SP)
5537  026674  012705  000021                    MOV     #17.,R5         ;2 ECC WORDS
5538                                                                    ;1 DATA GAP
5539                                                                    ;14 TOLERANCE GAP
5540  026700  005020                    2$:     CLR     (R0)+           ;CLEAR ECC, DATA GAP, AND
5541  026702  005305                            DEC     R5              ;TOLERANCE GAP
5542  026704  001375                            BNE     2$              ;BRANCH IF NOT COMPLETE
5543
5544
5545                                    ;*THESE ARE TO SETUP FOR DISKLESS USE ONLY
5546
5547  026706  012737  010000  047200            MOV     #01FMT22,@#CYL ;16 BITS PER WORD
5548                                                                    ;CYLINDER 0, FORMAT 16 BITS
5549  026714  112737  000000  047203            MOVB    #0,@#SECOTR+1  ;TRACK 0
5550  026722  112737  000001  047202            MOVB    #1,@#SECOTR    ;SECTOR 1
5551  026730  012737  000000  047204            MOV     #0,@#KEY1      ;KEY1=0
5552  026736  012737  000000  047206            MOV     #0,@#KEY2      ;KEY2=0
5553  026744  012737  000400  047260            MOV     #256.,@#DAWORD ;NO. OF DATA WORDS
5554  026752  005037  047210                    CLR     @#X            ;THIS IS A READ COMMAND
5555  026756  004537  043742                    JSR     R5,@#CRC       ;GO TO CALCULATE CRC
5556  026762  047200                            CYL
5557  026764  051100                            WCRC
5558
5559                                    ;*THESE ARE REGULAR SETUPS FOR RH11 AND "REINTO"
5560
5561  026766  004737  042534                    JSR     PC,@#CLDISK     ;SETUP GENERAL REGISTERS
5562  026772  012777  177374  152632            MOV     #-256.-4.,@RHWC ;256. DATA 4 HEADER WORDS
5563  027000  012777  003154  152626            MOV     #REINTO,@RHBA   ;STARTING ADDRESS OF READ BUFFER
5564  027006  112746  000001                    MOVB    #1,-(SP)        ;IN LOWER BYTE GET SECTOR
5565  027012  112766  000000  000001            MOVB    #0,1(SP)        ;GET TRACK IN HIGHER BYTE
5566  027020  012677  152620                    MOV     (SP)+,@RHDST    ;TRACK/SECTOR IN RHDST
5567  027024  012777  014000  152616            MOV     #FMT22!ECI,@RHOF ;16 BITS PER WORD
5568                                                                    ;ECC CORRECTION INHIBIT
```

H11

```
5569                                                    ;BECAUSE ECC IS NOT GOING
5570                                                    ;TO BE CHECKED
5571  027032  005077  152614          CLR     @RHCA     ;CYLINDER 0
5572
5573  027036  004737  042570          JSR     PC,@#CHECKT     ;CHECK THAT DVA,RDY,DPR,DRY = 1
5574  027042  104401  062145          TYPE    ,CPHALT         ;AND THAT NO OTHERS = 1.  CANNOT CON-
5575                                                    ;TINUE TESTING IF BOTH AREN'T TRUE
5576  027046  000000                  HALT                    ;STOP THE TEST
5577
5578  027050  013711  002070          MOV     @#REFOR,@R1     ;READ HEADER AND DATA=72
5579  027054  005037  002006          CLR     @#ERFLG$        ;CLEAR ERROR FLAG
5580  027060  004737  047040          JSR     PC,@#COMHD      ;READ HEADER AND DATA
5581
5582
5583                                  ;*IF THE PROGRAM COMES BACK HERE WITHOUT ERROR PRINTOUTS
5584                                  ;*FROM THE "COMHD" ROUTINE THAT MEANS SECTOR GAP,
5585                                  ;*FIRST SYNC, HEADER, HEADER CRC, HEADER GAP AND
5586                                  ;*SYNC BYTE HAVE GONE BY AND SYNCS WERE CORRECTLY
5587                                  ;*DETECTED.
5588
5589
5590                                  ;*RHWC IS CHECKED TO BE = 0 AFTER THE READ OPERATION
5591
5592  027064  017737  152542  001126  MOV     @RHWC,$PDDAT    ;LOAD AND TEST RHWC
5593  027072  001401                  BEQ     20$             ;SHOULD = 0
5594  027074  104040                  ERROR   40              ;RHWC DOES NOT = 0 AFTER A READ
5595                                                    ;HEADER AND DATA
5596
5597                                  ;*HEADER AND DATA ARE TO BE CHECKED.
5598                                  ;*IN CHECKING READ DATA THE WRITE FROM BUFFER
5599                                  ;*"WRFROM" IS FILLED WITH EXPECTED DATA AND
5600                                  ;*COMPARISONS ARE MADE.
5601
5602  027076  005737  002006    20$:  TST     @#ERFLGS        ;ANY ERRORS ALREADY THERE
5603  027102  001046                  BNE     TST64           ;BRANCH IF YES
5604  027104  004737  042760          JSR     PC,@#CHECKE     ;CHECK THAT BITS = 1
5605  027110  104401  062145          TYPE    ,CPHALT         ;CANNOT CONTINUE TESTING IF THEY DON'T
5606  027114  000000                  HALT                    ;STOP THE TEST
5607  027116  012700  002110          MOV     #WRFROM,R0      ;GETTING READY TO FILL EXPECTED DATA
5608  027122  012720  010000          MOV     #01FMT22,(R0)+  ;CYLINDER 0
5609  027126  112746  000001          MOVB    #1,-(SP)        ;IN LOWER BYTE GET SECTOR
5610  027132  112766  000000  000001  MOVB    #0,1(SP)        ;GET TRACK IN HIGHER BYTE
5611  027140  012620                  MOV     (SP)+,(R0)+     ;GET TRACK/SECTOR IN BUFFER
5612  027142  012720  000000          MOV     #0,(R0)+        ;KEY1 IN BUFFER
5613  027146  012720  000000          MOV     #0,(R0)+        ;KEY2 IN BUFFER
5614  027152  012701  000400          MOV     #256.,R1        ;DATA WORD COUNTER
5615  027156  012702  177777          MOV     #-1,R2          ;DATA
5616  027162  010220              3$: MOV     R2,(R0)+        ;DATA INTO BUFFER
5617  027164  005301                  DEC     R1              ;COUNT
5618  027166  001375                  BNE     3$              ;BRANCH IF 256 NOT DONE
5619
5620                                  ;*NOW READ DATA BUFFER WILL BE CHECKED
5621
5622  027170  004037  043430          JSR     R0,@#COMPAR     ;CHECK
5623  027174  002110                  WRFROM                  ;GOOD BUFFER
5624  027176  003154                  REINTO                  ;TEST BUFFER
```

```
5625  027200  000404                    4+256.                ;NUMBER OF WORDS CHECKED
5626  027202  027210                    4$                    ;RETURN POINT FOR ERROR HEADER
5627  027204  027214                    5$                    ;RETURN POINT FOR ERROR DATA
5628  027206  027220                    TST64                 ;RETURN FOR GOOD COMPARISON
5629  027210  104004          4$:       ERROR   4             ;READ NEXT ERROR
5630  027212  000207                    RTS     PC            ;RETURN TO "COMPAR"
5631  027214  104005          5$:       ERROR   5             ;WORD NOS 1 TO 4 ARE
5632                                                          ;HEADER WORDS
5633                                                          ;5 TO 260 ARE DATA WORDS
5634  027216  000207                    RTS     PC            ;RETURN TO "COMPAR"
5635
5636
5637
5638
```

```
5639
5640                                    ;;*****************************************************************
5641                                    ;*TEST 64       READ HEADER AND DATA 3
5642                                    ;**     READ CYLINDER 0 FORMAT 16 BITS PER WORD
5643                                    ;**     TRACK 1, SECTOR 1, KEYS 0, 256 WORDS OF 052525
5644                                    ;**     ANY ERROR LOGIC INDICATIONS ARE NOT CONCLUSIVE ON FIRST PASS
5645                                    ;**     BECAUSE ERROR LOGIC HAS NOT BEEN CHECKED YET (ON FIRST PASS)
5646
5647                                    ;;*****************************************************************
5648  027220  000004                    TST64:  SCOPE
5649  027222  012706  001000                    MOV     #STACK,SP       ;RESET STACK
5650  027226  012737  000064  002032            MOV     #TTNO,@#TSTNM   ;THIS SAVES TEST NUMBER
5651
5652
5653                                    ;*      SETUP FOR WHAT IS TO BE READ
5654                                    ;*      HEADER CRC IS RESTORED FROM A SUBROUTINE
5655
5656  027234  012746  052525                    MOV     #052525,-(SP)   ;DATA TO BE READ
5657  027240  012705  000400                    MOV     #256.,R5        ;COUNTER
5658  027244  012700  051116                    MOV     #DISK,R0        ;START OF SIMULATED DISK DATA
5659  027250  011620                    1$:     MOV     (SP),(R0)+      ;MOVE IN DATA ON TO SIMULATED DISK
5660  027252  005305                            DEC     R5              ;COUNT
5661  027254  001375                            BNE     1$              ;BRANCH IF 256 NOT COMPLETE
5662  027256  005726                            TST     (SP)+           ;UNDO -(SP)
5663  027260  012705  000021                    MOV     #17.,R5         ;2 ECC WORDS
5664                                                                    ;1 DATA GAP
5665                                                                    ;14 TOLERANCE GAP
5666  027264  005020                    2$:     CLR     (R0)+           ;CLEAR ECC, DATA GAP, AND
5667  027266  005305                            DEC     R5              ;TOLERANCE GAP
5668  027270  001375                            BNE     2$              ;BRANCH IF NOT COMPLETE
5669
5670
5671                                    ;*THESE ARE TO SETUP FOR DISKLESS USE ONLY
5672
5673  027272  012737  010000  047200            MOV     #01FMT22,@#CYL ;16 BITS PER WORD
5674                                                                    ;CYLINDER 0, FORMAT 16 BITS
5675  027300  112737  000001  047203            MOVB    #1,@#SECOTR+1  ;TRACK 1
5676  027306  112737  000001  047202            MOVB    #1,@#SECOTR    ;SECTOR 1
5677  027314  012737  000000  047204            MOV     #0,@#KEY1      ;KEY1=0
5678  027322  012737  000000  047206            MOV     #0,@#KEY2      ;KEY2=0
5679  027330  012737  000400  047260            MOV     #256.,@#DAWORD  ;NO. OF DATA WORDS
5680  027336  005037  047210                    CLR     @#X             ;THIS IS A READ COMMAND
5681  027342  004537  043742                    JSR     R5,@#CRC        ;GO TO CALCULATE CRC
5682  027346  047200                            CYL
5683  027350  051100                            WCRC
5684
5685                                    ;*THESE ARE REGULAR SETUPS FOR RH11 AND "REINTO"
5686
5687  027352  004737  042534                    JSR     PC,@#CLDISK     ;SETUP GENERAL REGISTERS
5688  027356  012777  177374  152246            MOV     #-256.-4.,@RHWC ;256. DATA 4 HEADER WORDS
5689  027364  012777  003154  152242            MOV     #REINTO,@RHBA   ;STARTING ADDRESS OF READ BUFFER
5690  027372  112746  000001                    MOVB    #1,-(SP)        ;IN LOWER BYTE GET SECTOR
5691  027376  112766  000001  000001            MOVB    #1,1(SP)        ;GET TRACK IN HIGHER BYTE
5692  027404  012677  152234                    MOV     (SP)+,@RHDST    ;TRACK/SECTOR IN RHDST
5693  027410  012777  014000  152232            MOV     #FMT22!ECI,@RHOF ;16 BITS PER WORD
5694                                                                    ;ECC CORRECTION INHIBIT
```

KII

```
5695
5696                                                    ;BECAUSE ECC IS NOT GOING
5697                                                    ;TO BE CHECKED
5697  027416  005077  152230          CLR     @RHCA     ;CYLINDER 0
5698
5699  027422  004737  042570          JSR     PC,@#CHECKT     ;CHECK THAT DVA,RDY,DPR,DRY = 1
5700  027426  104401  062145          TYPE    ,CPHALT         ;AND THAT NO OTHERS = 1.  CANNOT CON-
5701                                                    ;TINUE TESTING IF BOTH AREN'T TRUE
5702  027432  000000                  HALT              ;STOP THE TEST
5703
5704  027434  013711  002070          MOV     @#REFOR,@R1     ;READ HEADER AND DATA=72
5705  027440  005037  002006          CLR     @#ERFLG$        ;CLEAR ERROR FLAG
5706  027444  004737  047040          JSR     PC,@#COMHD      ;READ HEADER AND DATA
5707
5708
5709                                    ;*IF THE PROGRAM COMES BACK HERE WITHOUT ERROR PRINTOUTS
5710                                    ;*FROM THE "COMHD" ROUTINE THAT MEANS SECTOR GAP,
5711                                    ;*FIRST SYNC, HEADER, HEADER CRC, HEADER GAP AND
5712                                    ;*SYNC BYTE HAVE GONE BY AND SYNCS WERE CORRECTLY
5713                                    ;*DETECTED.
5714
5715
5716                                    ;*RHWC IS CHECKED TO BE = 0 AFTER THE READ OPERATION
5717
5718  027450  017737  152156  001126  MOV     @RHWC,$BDDAT    ;LOAD AND TEST RHWC
5719  027456  001401                  BEQ     20$             ;SHOULD = 0
5720  027460  104040                  ERROR   40              ;RHWC DOES NOT = 0 AFTER A READ
5721                                                    ;HEADER AND DATA
5722
5723                                    ;*HEADER AND DATA ARE TO BE CHECKED.
5724                                    ;*IN CHECKING READ DATA THE WRITE FROM BUFFER
5725                                    ;*"WRFROM" IS FILLED WITH EXPECTED DATA AND
5726                                    ;*COMPARISONS ARE MADE.
5727
5728  027462  005737  002006    20$:  TST     @#ERFLG$        ;ANY ERRORS ALREADY THERE
5729  027466  001046                  BNE     TST65           ;BRANCH IF YES
5730  027470  004737  042760          JSR     PC,@#CHECKE     ;CHECK THAT BITS = 1
5731  027474  104401  062145          TYPE    ,CPHALT         ;CANNOT CONTINUE TESTING IF THEY DON'T
5732  027500  000000                  HALT              ;STOP THE TEST
5733  027502  012700  002110          MOV     #WRFROM,R0      ;GETTING READY TO FILL EXPECTED DATA
5734  027506  012720  010000          MOV     #01!PMT22,(R0)+ ;CYLINDER 0
5735  027512  112746  000001          MOVB    #1,-(SP)        ;IN LOWER BYTE GET SECTOR
5736  027516  112766  000001  000001  MOVB    #1,1(SP)        ;GET TRACK IN HIGHER BYTE
5737  027524  012620                  MOV     (SP)+,(R0)+     ;GET TRACK/SECTOR IN BUFFER
5738  027526  012720  000000          MOV     #0,(R0)+        ;KEY1 IN BUFFER
5739  027532  012720  000000          MOV     #0,(R0)+        ;KEY2 IN BUFFER
5740  027536  012701  000400          MOV     #256.,R1        ;DATA WORD COUNTER
5741  027542  012702  052525          MOV     #052525,R2              ;DATA
5742  027546  010220              3$: MOV     R2,(R0)+        ;DATA INTO BUFFER
5743  027550  005301                  DEC     R1              ;COUNT
5744  027552  001375                  BNE     3$              ;BRANCH IF 256 NOT DONE
5745
5746                                    ;*NOW READ DATA BUFFER WILL BE CHECKED
5747
5748  027554  004037  043430          JSR     R0,@#COMPAR     ;CHECK
5749  027560  002110                  WRFROM                  ;GOOD BUFFER
5750  027562  003154                  REINTO                  ;TEST BUFFER
```

```
5751  027564  000404                    4+256.                ;NUMBER OF WORDS CHECKED
5752  027566  027574                    4$                    ;RETURN POINT FOR ERROR HEADER
5753  027570  027600                    5$                    ;RETURN POINT FOR ERROR DATA
5754  027572  027604                    TST65                 ;RETURN FOR GOOD COMPARISON
5755  027574  104004              4$:   ERROR   4             ;READ NEXT ERROR
5756  027576  000207                    RTS     PC            ;RETURN TO "COMPAR"
5757  027600  104005              5$:   ERROR   5             ;WORD NOS 1 TO 4 ARE
5758                                                          ;HEADER WORDS
5759                                                          ;5 TO 260 ARE DATA WORDS
5760  027602  000207                    RTS     PC            ;RETURN TO "COMPAR"
5761
5762
5763
5764
```

```
                                ;;*****************************************************************
                                ;*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
                                ;
                                ;;*************************************************************
027604  000004                TST65:  SCOPE
                                ;
027606  012737  000065  002032        MOV     #TTNO,@#TSTNM   ;THIS SAVES TEST NUMBER
                                ;
                                ;
027614  012706  001000        MOV     #STACK,SP       ;RESET STACK
027620  012737  000065  002032        MOV     #TTNO,@#TSTNM   ;THIS SAVES TEST NUMBER
                                ;
027626  004037  042452        JSR     R0,@#CLAREA     ;CLEAR SIMULATED DISK
027632  051116                .WORD   DISK            ;FROM
027634  052142                .WORD   TOLGAP+16       ;TO
027636  000000                .WORD   0               ;DATA
                                ;
                                ;*THESE ARE SETUP FOR DISKLESS USE ONLY
                                ;
027640  012737  010000  047200        MOV     #01FMT22,@#CYL;CYLINDER 0
                                                        ;16 BITS PER WORD
027646  112737  000000  047203        MOVB    #0,@#SECOTR+1;TRACK 0
027654  112737  000000  047202        MOVB    #0,@#SECOTR     ;SECTOR 0
027662  005037  047204        CLR     @#KEY1          ;KEY1 0
027666  005037  047206        CLR     @#KEY2          ;KEY2 0
027672  012737  000400  047246        MOV     #256.,@#NOWORD  ;NO OF DATA WORDS
027700  012737  000001  047210        MOV     #1,@#X          ;WRITE DATA
027706  004537  043742        JSR     R5,@#CRC        ;GO TO CALCULATE CRC
027712  047200                CYL
027714  051100                WCRC
                                ;
                                ;*THESE ARE REGULAR SETUPS
                                ;
027716  004037  042452        JSR     R0,@#CLAREA     ;FILL WRITE BUFFER WITH 377
027722  002110                WRFROM                  ;FROM LOCATION
027724  003110                WRFROM+<256.*2>         ;TO LOCATION
027726  000377                377                     ;DATA
027730  004737  042534        JSR     PC,@#CLDISK     ;SETUP GENERAL REGISTERS
027734  012777  177400  151670        MOV     #-256.,@RHWC    ;256. DATA WORDS
027742  012777  002110  151664        MOV     #WRFROM,@RHBA   ;STARTING ADDRESS OF WRITE BUFFER
027750  012746  000000        MOV     #0,-(SP)        ;SECTOR 0
027754  112766  000000  000001        MOVB    #0,1(SP)        ;TRACK 0
027762  012677  151656        MOV     (SP)+,@RHDST    ;SECTOR 0 TRACK 0
027766  012777  010000  151654        MOV     #FMT22,@RHOF    ;16 BITS PER WORD FORMAT
027774  012777  000000  151650        MOV     #0,@RHCA        ;CYLINDER 0
030002  004737  042570        JSR     PC,@#CHECKT     ;CHECK THAT DVA,RDY,DPR,DRY = 1
030006  104401  062145        TYPE    ,CPHALT         ;AND THAT NO OTHERS = 1.  CANNOT CON-
                                                        ;TINUE TESTING IF BOTH AREN'T TRUE
030012  000000                HALT                    ;STOP THE TEST
030014  013711  002062        MOV     @#WRIDAT,@R1    ;WRITE DATA=60
```

A12

```
5821  030020  005037  002006                  CLR     @#ERFLG$        ;CLEAR ERROR FLAG
5822  030024  004737  047040                  JSR     PC,@#COMHD      ;WRITE DATA
5823
5824
5825                                  ;*IF THE PROGRAM COMES BACK HERE WITHOUT ERROR PRINTOUTS
5826                                  ;*FROM THE "COMHD" ROUTINE IT MEANS SECTOR GAP, SYNC BYTE
5827                                  ;*HEADER, HEADER CRC, HEADER GAP AND SYNC BYTE HAVE GONE BY
5828                                  ;*AND SYNCS WERE CORRECTLY DETECTED, DATA IS TO BE CHECKED.
5829
5830  030030  004737  042140                  JSR     PC,@#PUTREG     ;SAVE REGISTERS
5831  030034  005737  002006                  TST     @#ERFLG$        ;HAVE ANY ERRORS OCCURED?
5832  030040  001041                          BNE     TST66           ;BRANCH IF YES
5833  030042  012700  000377                  MOV     #377,R0         ;GOOD DATA
5834  030046  012701  051116                  MOV     #DISK,R1        ;DATA WRITTEN INTO "DISK"
5835  030052  012702  000400                  MOV     #256.,R2        ;COUNTER
5836
5837  030056  012737  000401  047320  1$:     MOV     #256.+1,@#ERWORD;FOR ERROR WORD
5838  030064  020021                          CMP     R0,(R1)+        ;COMPARE GOOD DATA WITH DATA ON DISK
5839  030066  001424                          BEQ     3$              ;BRANCH IF GOOD
5840  030070  010037  001124                  MOV     R0,@#$GDDAT     ;GOOD DATA
5841  030074  014137  001126                  MOV     -(R1),@#$BDDAT  ;BAD DATA
5842  030100  160237  047320                  SUB     R2,@#ERWORD     ;ERROR WORD NO
5843  030104  005737  002006                  TST     @#ERFLG$        ;ANY ERRORS ALREADY THERE?
5844  030110  001002                          BNE     2$              ;BRANCH IF YES
5845  030112  104004                          ERROR   4               ;ERROR ON WRITE DATA COMMAND
5846  030114  000401                          BR      64$             ;BRANCH TO AVOID PRINTING NEXT ERROR
5847
5848  030116  104005                  2$:     ERROR   5               ;WORD NO GIVES WORD IN ERROR
5849  030120  005721                  64$:    TST     (R1)+           ;UNDO -(R1) FOR BAD DATA
5850  030122  017746  151012                  MOV     @SWR,-(SP)      ;GET SWITCH SETTING
5851  030126  042716  177177                  BIC     #177177,(SP)    ;KEEP ONLY SWITCH 7 AND 8
5852  030132  022726  000200                  CMP     #SW07,(SP)+     ;IS 7 SET AND 8 RESET
5853  030136  001402                          BEQ     TST66           ;BRANCH OUT IF YES
5854  030140  005302                  3$:     DEC     R2              ;IF NOT COUNT 256 WORDS
5855  030142  001345                          BNE     1$              ;BRANCH IF 256. NOT DONE
5856
5857
5858
5859
```

```
                                ;;*****************************************************************
                                ;*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)
                                
                                ;;*****************************************************************
030144  000004                  TST66:  SCOPE
030146  012706  001000                  MOV     #STACK,SP       ;RESET STACK
030152  012737  000066  002032          MOV     #TTNO,@#TSTNM   ;THIS SAVES TEST NUMBER
030160  004037  042452                  JSR     R0,@#CLAREA     ;CLEAR SIMULATED DISK
030164  051116                          .WORD   DISK            ;FROM
030166  052114                          .WORD   DISK+776        ;TO
030170  177400                          .WORD   177400          ;DATA
030172  004037  042452                  JSR     R0,@#CLAREA     ;CLEAR READ INTO BUFFER
030176  003154                          .WORD   REINTO          ;FROM
030200  004152                          .WORD   REINTO+776      ;TO
030202  000000                          .WORD   0               ;DATA
                                
                                        ;*THESE ARE TO SETUP FOR DISKLESS USE ONLY
                                
030204  012737  010000  047200          MOV     #FMT22,@#CYL    ;CYLINDER 0 16 BITS PER WORD FORMAT
030212  105037  047203                  CLRB    @#SECOTR+1      ;TRACK 0
030216  112737  000001  047202          MOVB    #1,@#SECOTR     ;SECTOR 1
030224  005037  047204                  CLR     @#KEY1          ;KEY1=0
030230  005037  047206                  CLR     @#KEY2          ;KEY2=0
030234  012737  000012  047260          MOV     #10.,@#DAWORD   ;NO. OF DATA WORDS
030242  005037  047210                  CLR     @#X             ;THIS IS A READ COMMAND
030246  004537  043742                  JSR     R5,@#CRC        ;GO TO CALCULATE CRC
030252  047200                          CYL
030254  051100                          WCRC
                                
                                        ;*THESE ARE REGULAR SETUPS
                                
030256  004737  042534                  JSR     PC,@#CLDISK     ;SETUP GENERAL REGISTERS
030262  013711  002066                  MOV     @#READAT,@R1    ;READ DATA INTO RHCS1=70
030266  012777  177766  151336          MOV     #-10.,@RHWC     ;10 DATA WORDS
030274  012777  003154  151332          MOV     #REINTO,@RHBA   ;STARTING ADDRESS OF READ BUFFER
030302  112746  000001                  MOVB    #1,-(SP)        ;IN LOWER BYTE GET SECTOR 1
030306  112766  000000  000001          MOVB    #0,1(SP)        ;GET TRACK0 IN UPPER BYTE
030314  012677  151324                  MOV     (SP)+,@RHDST    ;TRACK/SECTOR IN RHDST
030320  012777  014000  151322          MOV     #FMT22!ECI,@RHOF ;16 BITS PER WORD
                                                                ;ECC CORRECTION INHIBIT BECAUSE
                                                                ;ECC IS NOT CHECKED HERE
030326  005077  151320                  CLR     @RHCA           ;CYLINDER 0
030332  004737  042570                  JSR     PC,@#CHECKT     ;CHECK THAT DVA,RDY,DPR,DRY = 1
030336  104401  062145                  TYPE    ,CPHALT         ;AND THAT NO OTHERS = 1.  CANNOT CON-
                                                                ;TINUE TESTING IF BOTH AREN'T TRUE
030342  000000                          HALT                    ;STOP THE TEST
030344  005037  002006                  CLR     @#ERFLG$        ;CLEAR ERROR FLAG
030350  004737  047040                  JSR     PC,@#COMND      ;READ DATA
```

C12

```
5916
5917                                    ;*IF THE PROGRAM COMES BACK HERE WITHOUT ERROR PRINTOUT
5918                                    ;*FROM "COMHD" ROUTINE IN MEANS DATA IS TO BE CHECKED
5919
5920                                    ;*NOW THE DATA READ INTO "REINTO" BUFFER WILL
5921                                    ;*BE CHECKED, ONLY 10 WORDS SHOULD BE CHANGED
5922                                    ;*ALL OTHER WORDS SHOULD REMAIN UNCHANGED
5923                                    ;*THE "WRFROM" BUFFER IS FILLED WITH EXPECTED DATA AND CHECKED
5924
5925  030354  005737  002006          TST     @#ERFLG$        ;HAVE ANY ERRORS OCCURED?
5926  030360  001053                  BNE     TST67           ;;BRANCH IF YES
5927  030362  004037  042452          JSR     R0,@#CLAREA     ;CLEAR BUFFER
5928  030366  002110                  WRFROM                  ;FROM
5929  030370  003106                  WRFROM+776              ;TO
5930  030372  000000                  0                       ;DATA
5931
5932  030374  004037  042452          JSR     R0,@#CLAREA     ;FILL EXPECTED DATA
5933  030400  002110                  WRFROM                  ;FROM
5934  030402  002132                  WRFROM+22               ;TO
5935  030404  177400                  177400                  ;DATA
5936
5937                                    ;*NOW READ DATA BUFFER IS CHECKED
5938
5939  030406  012700  002110          MOV     #WRFROM,R0      ;GOOD DATA
5940  030412  012701  003154          MOV     #REINTO,R1      ;DATA READ
5941  030416  012702  000400          MOV     #256.,R2        ;COUNTER
5942  030422  012737  000401  047320  1$:     MOV     #257.,@#ERWORD  ;FOR ERROR WORD NO
5943  030430  022021                  CMP     (R0)+,(R1)+     ;COMPARE GOOD WITH READ BUFFER
5944  030432  001424                  BEQ     2$              ;BRANCH IF GOOD
5945  030434  014037  001124          MOV     -(R0),@#$GDDAT  ;GOOD DATA
5946  030440  014137  001126          MOV     -(R1),@#$BDDAT  ;BAD DATA
5947  030444  160237  047320          SUB     R2,@#ERWORD     ;ERROR WORD NO
5948  030450  005737  002006          TST     @#ERFLG$        ;ANY ERRORS ALREADY THERE
5949  030454  001002                  BNE     3$              ;IF YES BRANCH DO NOT TYPE HEADER
5950  030456  104004                  ERROR   4               ;ERROR ON READ DATA
5951  030460  000401                  BR      4$              ;BRANCH TO AVOID PRINTING NEXT ERROR
5952  030462  104005          3$:     ERROR   5               ;WORD NO 1-10 ARE DATA
5953                                                          ;WORDS
5954                                                          ;WORD NOS 11-256 HAVE NOT BEEN
5955                                                          ;READ AND BUFFER SHOULD BE
5956                                                          ;ZERO IF OTHER THAN ZERO
5957                                                          ;WRONG NUMBER OF WORDS HAVE
5958                                                          ;BEEN READ IN THE DISK NOW
5959                                                          ;CONTAINS 177400 ALL 256
5960                                                          ;WORDS BUT ONLY 10 WORDS
5961                                                          ;SHOULD BE READ IN
5962
5963  030464  022021          4$:     CMP     (R0)+,(R1)+     ;UNDO -(R0) AND -(R1) FOR ERROR
5964  030466  017746  150446          MOV     @SWR,-(SP)      ;GET SWITCH SETTING
5965  030472  042716  177177          BIC     #177177,(SP)    ;KEEP ONLY SWITCH 7 AND 8
5966  030476  022726  000200          CMP     #SW07,(SP)+     ;IS 7 SET AND 8 RESET
5967  030502  001402                  BEQ     TST67           ;BRANCH OUT IF YES
5968  030504  005302          2$:     DEC     R2              ;COUNT
5969  030506  001345                  BNE     1$              ;BRANCH IF NOT COMPLETE
5970
5971
```

```
 5972
```

```
                                ;*
                                ;;*****************************************************************
                                ;*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
                                ;;*****************************************************************
030510  000004                  TST67:  SCOPE

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

030512  012706  001000                  MOV     #STACK,SP       ;RESET STACK
030516  012737  000067  002032          MOV     #TTWO,@#TSTNM   ;THIS SAVES TEST NUMBER

                                ;*SET UP "REINTO" FOR WHAT IS TO BE READ
030524  012701  003154                  MOV     #REINTO,R1      ;STARTING ADDRESS
030530  012721  010000                  MOV     #FMT22,(R1)+    ;CYLINDER 0 FORMAT 16 BIT WORDS
030534  012721  000401                  MOV     #401,(R1)+      ;TRACK=1, SECTOR=1
030540  005021                          CLR     (R1)+           ;KEY1=0
030542  005021                          CLR     (R1)+           ;KEY2=0
030544  004037  042452                  JSR     R0,@#CLAREA     ;FILL "REINTO" BUFFER
030550  003164                          .WORD   REINTO+<4*2>    ;FROM
030552  003232                          .WORD   REINTO+<23.*2>  ;TO
030554  070707                          .WORD   070707          ;DATA

030556  012700  177776                  MOV     #177776,R0      ;GETTING READY TO FLOAT 0
030562  012701  003234                  MOV     #REINTO+<24.*2>,R1;STARTING ADDRESS WHERE 177776 GOES
030566  010021          1$:     MOV     R0,(R1)+        ;MOVE IN FLOATING 0
030570  000261                  SEC                     ;SET CARRY
030572  006100                  ROL     R0              ;GET 0 ONE BIT LEFT
030574  103774                  BCS     1$              ;BRANCH IF 16 NOT DONE

030576  004037  042452                  JSR     R0,@#CLAREA     ;FILL THE REST OF BUFFER WITH 0
030602  003274                          .WORD   REINTO+<40.*2>  ;FROM
030604  004152                          .WORD   REINTO+776      ;TO
030606  000000                          .WORD   0               ;DATA

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

030610  004037  042452                  JSR     R0,@#CLAREA     ;FILL "DISK" BUFFER
030614  051116                          .WORD   DISK            ;FROM
030616  051164                          .WORD   DISK+<19.*2>    ;TO
030620  070707                          .WORD   070707          ;DATA

030622  012700  177776                  MOV     #177776,R0      ;GETTING READY TO FLOAT ZEROS
030626  012701  051166                  MOV     #DISK+<20.*2>,R1;STARTING ADDRESS WHERE 177776 GOES
030632  010021          2$:     MOV     R0,(R1)+        ;MOVE IN FLOATING 0
030634  000261                  SEC                     ;SET CARRY
```

F12

```
6029  030636  006100                      ROL     R0              ;GET 0 ONE BIT LEFT
6030  030640  103774                      BCS     2$              ;BRANCH IF 16 NOT DONE
6031
6032  030642  004037  042452              JSR     R0,@#CLAREA     ;FILL THE REST OF BUFFER WITH 177777
6033  030646  051226                      .WORD   DISK+<36.*2>    ;FROM
6034  030650  052114                      .WORD   DISK+776        ;TO
6035  030652  177777                      .WORD   177777          ;DATA
6036
6037  030654  004737  043260              JSR     PC,@#WRCHHD     ;WRITE CHECK HEADER AND DATA
6038                                                              ;CYLINDER 0, TRACK 1, SECTOR 1
5039
6040                                      ;*IF THE PROGRAM COMES BACK HERE THEN WRITE CHECK
6041                                      ;*HAS BEEN COMPLETED NOW WRITE CHECK ERROR BIT IS TO BE TESTED
6042
6043  030660  013746  001774              MOV     @#UNIT,-(SP)    ;GET UNIT NUMBER
6044  030664  052716  000100              BIS     #IR,(SP)        ;ONLY BIT 6 SHOULD BE SET
6045  030670  004737  042140              JSR     PC,@#PUTREG     ;SAVE REGISTERS
6046  030674  022637  001712              CMP     (SP)+,@#CS2     ;COMPARE RHCS2
6047  030700  001406                      BEQ     4$              ;BRANCH IF GOOD
6048  030702  032712  040000              BIT     #WCE,@R2        ;WRITE CHECK ERROR HIGH?
6049  030706  001402                      BEQ     3$              ;BRANCH IF ERROR NOT DUE TO "WCE"
6050  030710  104017                      ERROR   17              ;RHDB CONTAINS FAILING WORD
6051  030712  000401                      BR      4$              ;RHBA CONTAINS ADDRESS+2
6052                                                              ;OF THE WORD IN MEMORY FROM
6053                                                              ;THE DISK THAT DID NOT COMPARE
6054                                                              ;TRE AND SC WILL BE SET DUE TO
6055                                                              ;WCE
6056  030714  104017              3$:     ERROR   17              ;WCE CORRECTLY WAS NOT SET BUT SOME
6057                                                              ;BITS OTHER THAN IR
6058                                                              ;AND UNIT NO. WAS SET
6059
6060                                      ;*NOW CHECK MEMORY TO SEE IF NOTHING GOT DESTROYED
6061                                      ;*FILL "WRFROM" WITH WHAT SHOULD BE IN "REINTO" THEN CHECK
6062
6063  030716  012700  002110      4$:     MOV     #WRFROM,R0      ;STARTING ADDRESS
6064  030722  012720  010000              MOV     #PAT22,(R0)+    ;CYLINDER
6065  030726  012720  000401              MOV     #401,(R0)+      ;TRACK=1, SECTOR=1
6066  030732  005020                      CLR     (R0)+           ;KEY1=0
6067  030734  005020                      CLR     (R0)+           ;KEY2=0
6068
6069  030736  004037  042452              JSR     R0,@#CLAREA     ;FILL "WRFROM" BUFFER
6070  030742  002120                      .WORD   WRFROM+<4*2>    ;FROM
6071  030744  002166                      .WORD   WRFROM+<23.*2>  ;TO
6072  030746  070707                      .WORD   070707          ;DATA
6073
6074  030750  012700  177776              MOV     #177776,R0      ;GETTING READY TO FLOAT 0
6075  030754  012701  002170              MOV     #WRFROM+<24.*2>,R1 ;STARTING ADDRESS WHERE 177776 GOES
6076  030760  010021              5$:     MOV     R0,(R1)+        ;MOVE IN FLOATING 0
6077  030762  000261                      SEC                     ;SET CARRY
6078  030764  006100                      ROL     R0              ;GET 0 ONE BIT LEFT
6079  030766  103774                      BCS     5$              ;BRANCH IF 16 NOT DONE
6080
6081  030770  004037  042452              JSR     R0,@#CLAREA     ;FILL THE REST OF BUFFER WITH 0
6082  030774  002230                      .WORD   WRFROM+<40.*2>  ;FROM
6083  030776  003106                      .WORD   WRFROM+776      ;TO
6084  031000  000000                      .WORD   0               ;DATA
```

GR

```
6085
6086                                                ;*NOW THE READ BUFFER WILL BE CHECKED
6087  031002  005037  002006                        CLR     @#ERFLG$        ;CLEAR ERROR FLAG
6088
6089  031006  004037  043430                        JSR     R0,@#COMPAR     ;CHECK
6090  031012  002110                                 WRFROM                  ;GOOD BUFFER
6091  031014  003154                                 REINTO                  ;TEST BUFFER
6092  031016  000400                                 256.                    ;NUMBER OF WORDS CHECKED
6093  031020  031026                                 6$                      ;RETURN POINT FOR ERROR HEADER
6094  031022  031032                                 7$                      ;RETURN POINT FOR ERROR DATA
6095  031024  031040                                 TST70                   ;RETURN FOR GOOD COMPARISON
6096  031026  104004                    6$:          ERROR   4               ;READ NEXT ERROR 5
6097  031030  000207                                 RTS     PC              ;RETURN TO COMPARISON SUBROUTINE
6098  031032  104005                    7$:          ERROR   5               ;DATA IN REINTO BUFFER GOT
6099                                                                          ;CHANGED AFTER A WRITE
6100                                                                          ;CHECK HEADER AND DATA COMMAND
6101                                                                          ;WORD NO CONTAINS THE WORD
6102                                                                          ;NUMBER THAT GOT CHANGED
6103  031034  000207                                 RTS     PC              ;RETURN TO COMPARISON SUBROUTINE
6104
6105  031036  000240                    10$:         NOP                     ;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

                        ;;*****************************************************************
031040  000004          TST70:  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)

031042  012706  001000          MOV     #STACK,SP       ;RESET STACK
031046  012737  000070  002032  MOV     #TTNO,@#TSTNM   ;THIS SAVES TEST NUMBER

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

031054  012700  000001          MOV     #1,R0           ;GETTING READY TO FLOAT 1
031060  012701  003154          MOV     #REINTO,R1      ;STARTING ADDRESS WHERE 1 GOES
031064  010021          1$:     MOV     R0,(R1)+        ;MOVE FLOATING 1
031066  006100          ROL     R0              ;GET 1 ONE BIT LEFT
031070  103375          BCC     1$              ;BRANCH IF 16 NOT DONE
031072  012700  177776          MOV     #177776,R0      ;GETTING READY TO FLOAT 0
031076  012701  003214          MOV     #REINTO+<16.*2>,R1 ;STARTING ADDRESS WHERE 177776 GOES
031102  010021          2$:     MOV     R0,(R1)+        ;MOVE IN FLOATING 0
031104  000261          SEC                     ;SET CARRY
031106  006100          ROL     R0              ;GET 0 ONE BIT LEFT
031110  103774          BCS     2$              ;BRANCH IF 16 NOT DONE

031112  004037  042452          JSR     R0,@#CLAREA     ;FILL REST OF BUFFER WITH 1
031116  003254          .WORD   REINTO+<32.*2>  ;FROM
031120  004152          .WORD   REINTO+776      ;TO
031122  000001          .WORD   1               ;WITH DATA

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

031124  012700  000001          MOV     #1,R0           ;GETTING READY TO FLOAT 1
031130  012701  051116          MOV     #DISK,R1        ;STARTING ADDRESS WHERE 1 GOES
031134  010021          3$:     MOV     R0,(R1)+        ;MOVE FLOATING 1
031136  006100          ROL     R0              ;GET 1 ONE BIT LEFT
031140  103375          BCC     3$              ;BRANCH IF 16 NOT DONE

031142  012700  177776          MOV     #177776,R0      ;GETTING READY TO FLOAT 0
031146  012701  051156          MOV     #DISK+<16.*2>,R1 ;STARTING ADDRESS WHERE 177776 GOES
031152  010021          4$:     MOV     R0,(R1)+        ;MOVE FLOATING 0
031154  000261          SEC                     ;SET CARRY
031156  006100          ROL     R0              ;GET 0 ONE BIT LEFT
031160  103774          BCS     4$              ;BRANCH IF 16 NOT DONE

```

```
6162  031162  004037  042452              JSR     R0,@#CLAREA     ;FILL REST OF BUFFER WITH 0
6163  031166  051216                      .WORD   DISK+<32.*2>    ;FROM
6164  031170  052114                      .WORD   DISK+776        ;TO
6165  031172  000000                      .WORD   0               ;WITH DATA
6166
6167  031174  004737  043572              JSR     PC,@#WRCHDA     ;WRITE CHECK DATA
6168                                                              ;CYLINDER 0, TRACK 1, SECTOR 1
6169                                                              ;KEYS 0, 32 WORDS.
6170
6171                                      ;*IF THE PROGRAM COMES BACK HERE THEN WRITE CHECK
6172                                      ;*HAS BEEN COMPLETED NOW WRITE CHECK ERROR BIT IS TESTED
6173
6174  031200  013746  001774              MOV     @#UNIT,-(SP)    ;GET UNIT NUMBER
6175  031204  052716  000100              BIS     #IR,(SP)        ;ONLY BIT 6 SHOULD BE SET
6176  031210  004737  042140              JSR     PC,@#PUTREG     ;SAVE REGISTERS
6177  031214  022637  001712              CMP     (SP)+,@#CS2     ;COMPARE RHCS2
6178  031220  001407                      BEQ     6$              ;BRANCH IF GOOD
6179  031222  032737  040000  001712      BIT     #WCE,@#CS2      ;WRITE CHECK ERROR HIGH?
6180  031230  001402                      BEQ     5$              ;BRANCH IF ERROR NOT DUE TO "WCE"
6181  031232  104017                      ERROR   17              ;RHDB CONTAINS FAILING WORD
6182  031234  000401                      BR      6$              ;RHBA CONTAINS ADDRESS+2
6183                                                              ;OF THE WORD IN MEMORY FROM
6184                                                              ;THE DISK THAT DID NOT COMPARE
6185                                                              ;TRE AND SC WILL BE SET DUE TO WCE
6186  031236  104017          5$:         ERROR   17              ;WCE WAS CORRECTLY NOT SET
6187                                                              ;BUT SOME BITS OTHER THAN
6188                                                              ;IR AND UNIT NO. WERE SET
6189
6190                                      ;*NOW CHECK MEMORY TO SEE IF ANYTHING GOT DESTROYED
6191                                      ;*FILL "WRFROM" WITH WHAT SHOULD BE IN REINTO THEN CHECK IT
6192
6193  031240  005037  002006      6$:     CLR     @#ERFLG$        ;CLEAR ERROR FLAG
6194  031244  012700  000001              MOV     #1,R0           ;GETTING READY TO FLOAT 1
6195  031250  012701  002110              MOV     #WRFROM,R1      ;START ADDRESS WHERE 1 GOES
6196  031254  010021          7$:         MOV     R0,(R1)+        ;MOVE FLOATING 1
6197  031256  006100                      ROL     R0              ;GET 1 ONE BIT LEFT
6198  031260  103375                      BCC     7$              ;BRANCH IF 16 NOT DONE
6199
6200  031262  012700  177776              MOV     #177776,R0      ;GETTING READY TO FLOAT 0
6201  031266  012701  002150              MOV     #WRFROM+<16.*2>,R1 ;STARTING ADDRESS WHERE 177776 GOES
6202  031272  010021          10$:        MOV     R0,(R1)+        ;MOVE IN FLOATING 0
6203  031274  000261                      SEC                     ;SET CARRY
6204  031276  006100                      ROL     R0              ;GET 0 ONE BIT LEFT
6205  031300  103774                      BCS     10$             ;BRANCH IF CARRY SET
6206
6207  031302  004037  042452              JSR     R0,@#CLAREA     ;FILL REST OF BUFFER WITH 1
6208  031306  002210                      .WORD   WRFROM+<32.*2>  ;FROM
6209  031310  003106                      .WORD   WRFROM+776      ;TO
6210  031312  000001                      .WORD   1               ;WITH DATA
6211
6212                                      ;*NOW THE READ BUFFER WILL BE CHECKED
6213
6214  031314  004037  043430              JSR     R0,@#COMPAR     ;CHECK
6215  031320  002110                      WRFROM                  ;GOOD BUFFER
6216  031322  003154                      REINTO                  ;TEST BUFFER
6217  031324  000400                      256.                    ;NUMBER OF WORDS CHECKED
```

```
6218  031326  031334                      11$                 ;RETURN POINT FOR ERROR HEADER
6219  031330  031340                      12$                 ;RETURN POINT FOR ERROR DATA
6220
6221  031332  031346                      TST71               ;RETURN FOR GOOD COMPARISON
6222
6223  031334  104004              11$:    ERROR   4           ;READ NEXT ERROR 5
6224  031336  000207                      RTS     PC          ;RETURN TO COMPARISON SUBROUTINE
6225  031340  104005              12$:    ERROR   5           ;DATA IN REINTO BUFFER GOT
6226                                                          ;CHANGED AFTER A WRITE
6227                                                          ;CHECK DATA COMMAND
6228                                                          ;WORD NO CONTAINS THE WORD
6229                                                          ;NUMBER THAT GOT CHANGED
6230  031342  000207                      RTS     PC          ;RETURN TO COMPARISON SUBROUTINE
6231
6232  031344  000240              13$:    NOP                 ;ONLY A BRANCH POINT
6233
6234
```

```
                                .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

                                ;;*******************************************************************
031346  000004                  TST71:  SCOPE

031350  012706  001000                  MOV     #STACK,SP       ;RESET STACK
031354  012737  000071  002032          MOV     #TTNO,@#TSTNM   ;THIS SAVES TEST NUMBER
031362  004737  042534                  JSR     PC,@#CLDISK     ;CLEAR DISK REGISTERS
031366  004737  042570                  JSR     PC,@#CHECKT     ;CHECK THAT DVA,RDY,DPR,DRY = 1
031372  104401  062145                  TYPE    ,CPHALT         ;AND THAT NO OTHERS = 1.  CANNOT CON-
                                                                ;TINUE TESTING IF BOTH AREN'T TRUE
031376  000000                          HALT                    ;STOP THE TEST



031400  012700  003154                  MOV     #REINTO,R0      ;BUFFER STARTING FOR 3 ERROR
                                                                ;REGISTERS
031404  013720  001642                  MOV     @#RHER1,(R0)+   ;RHER1 STORED IN REINTO
031410  012720  000000                  MOV     #0,(R0)+        ;BITS NOT TO BE CHECKED IN RHER1
031414  013720  001646                  MOV     @#RHER2,(R0)+   ;RHER2 STORED IN REINTO+4
031420  012720  000000                  MOV     #0,(R0)+        ;BITS NOT TO BE CHECKED IN RHER2
031424  013720  001654                  MOV     @#RHER3,(R0)+   ;RHER3 STORED IN REINTO+10
031430  012720  000000                  MOV     #0,(R0)+        ;BITS NOT TO BE CHECKED IN RHER3

031434  013704  001656                  MOV     @#RHAS,R4       ;R4 HAS RHAS
031440  013705  002016                  MOV     @#ATTENT,R5     ;R5 HAS ATA BIT IN RHAS
031444  012737  031472  001110          MOV     #2$,@#$LPERR    ;THAT SHOULD SET WITH ERROR
                                                                ;RETURN POINT TO ERROR
031452  012737  000003  001200          MOV     #3,@#$TMP1      ;ERROR REGISTER COUNTER
031460  012700  003154                  MOV     #REINTO,R0      ;REGISTER BUFFER POINTER

031464  012002                  1$:     MOV     (R0)+,R2        ;R2 HAS ADDRESS OF ERROR REG
031466  012701  000001                  MOV     #BIT0,R1        ;R1 WILL HAVE BIT UNDER TEST
031472  052777  000040  150136  2$:     BIS     #CLR,@RHCS2     ;CLEAR RHCS2
031500  013777  001774  150130          MOV     @#UNIT,@RHCS2   ;REINSTATE UNIT NO.
031506  010112                          MOV     R1,@R2          ;SET ERROR BIT
031510  004737  042140                  JSR     PC,@#PUTREG     ;READ AND SAVE REGISTERS
031514  120537  001732                  CMPB    R5,@#AS         ;ONLY THE BIT IN R5 SHOULD BE
                                                                ;SET IN RHAS
```

```
6291  031520  001401                    BEQ     3$              ;LOOK @ RHDS1 IF GOOD
6292  031522  104020                    ERROR   20              ;WITH THE SETTING OF ONE
6293                                                            ;ERROR BIT IN AN ERROR
6294                                                            ;REGISTER, THE CORRESPONDING
6295                                                            ;RHAS BIT DID NOT SET
6296
6297  031524  013746  001736      3$:   MOV     @#DS1,-(SP)     ;GET RHDS1
6298  031530  042716  001100            BIC     #VV!PROG,(SP)   ;REMOVE VV AND PROG
6299  031534  022726  140600            CMP     #ATA!ERR!DPR!DRY,(SP)+;THESE BITS PLUS VV SHOULD BE IN RHDS1
6300  031540  001401                    BEQ     4$              ;CHECK 'GO' NEXT, IF THIS WAS OK
6301  031542  104020                    ERROR   20              ;WITH THE SETTING OF ONE
6302                                                            ;ERROR BIT, COMPOSITE ERROR
6303                                                            ;OR ATTENTION ACTIVE, OR
6304                                                            ;ONE OF THE OTHER
6305                                                            ;PERMANENT BITS DID NOT SET
6306
6307  031544  012777  000001  150066  4$:   MOV     #GO,@RHCS1      ;GIVE NO-OP
6308  031552  004737  042140            JSR     PC,@#PUTREG     ;SAVE REGISTERS
6309  031556  020112                    CMP     R1,@R2          ;GO SHOULD NOT CLEAR ERROR
6310  031560  001410                    BEQ     5$              ;FURTHER CHECK OF 'GO' FUNCTIONALITY
6311  031562  010237  042204            MOV     R2,@#REGADR     ;FAILING REGISTER
6312  031566  010137  001124            MOV     R1,@#$GDDAT     ;GOOD DATA
6313  031572  013737  001712  001126    MOV     @#CS2,@#$BDDAT  ;BAD DATA
6314  031600  104001                    ERROR   1               ;"GO" WITH NO-OP CHANGED
6315                                                            ;ERROR REGISTER
6316
6317  031602  013746  001736      5$:   MOV     @#DS1,-(SP)     ;GET RHDS1
6318  031606  042716  001100            BIC     #VV!PROG,(SP)   ;CLEAR VV AND PROG
6319  031612  022726  140600            CMP     #ATA!ERR!DPR!DRY,(SP)+;GO SHOULD NOT CLEAR ANY BITS
6320  031616  001401                    BEQ     7$              ;CHECK NEXT ERROR BIT IF A-OK
6321  031620  104020                    ERROR   20              ;"GO" WITH NO-OP SHOULD NOT CLEAR
6322                                                            ;ATA AND/OR ERR
6323
6324
6325                              ;*THIS IS THE MAIN BIT TESTING CONTROL LOGIC
6326
6327
6328  031622  006301            7$:   ASL     R1              ;GET NEXT BIT TO THE LEFT
6329  031624  103403                    BCS     10$             ;GO ON TO NEXT REGISTER IF DONE
6330  031626  031001                    BIT     (R0),R1         ;IS THIS BIT TO BE TESTED ?
6331  031630  001374                    BNE     7$              ;IF NOT, GET NEXT ONE
6332  031632  000717                    BR      2$              ;IF TO BE TESTED, GO DO IT !
6333
6334  031634  005720            10$:  TST     (R0)+           ;ADVANCE R0 TO NEXT ERROR REG.
6335  031636  005337  001200            DEC     @#$TMP1         ;REGISTER COUNTER
6336  031642  001310                    BNE     1$              ;DO NEXT ONE, IF 3 NOT COMPLETE
6337
6338
6339                                    ;*NOW AFTER SETTING ATA IN RHDS1 "1" IN RHAS AT THE
6340                                    ;*DRIVE POSITION SHOULD CLEAR ATA IN RHDS1
6341
6342  031644  004737  042534      11$:  JSR     PC,@#CLDISK     ;CLEAR
6343  031650  012737  031644  001110    MOV     #11$,$LPERR     ;ERROR RETURN
6344  031656  012714  177777            MOV     #-1,@R4         ;SET BIT IN RHAS AND ATA IN RHDS1
6345  031662  013777  002016  147766    MOV     @#ATTENT,@RHAS  ;WRITE 1 INTO  DRIVE BIT POSITION
6346  031670  004737  042140            JSR     PC,@#PUTREG     ;SAVE REGISTERS
```

m12

```
6347  031674  105737  001732          TSTB    @#AS            ;THIS SHOULD BE ZERO
6348  031700  001401                  BEQ     12$
6349  031702  104020                  ERROR   20              ;MOVING A "1" INTO RHAS
6350                                                          ;AT THE DRIVE BIT POSITION
6351                                                          ;DID NOT CLEAR IT
6352
6353  031704  013746  001736  12$:    MOV     @#RDS1,-(SP)    ;GET RHDS1
6354  031710  042716  001000          BIC     #PROG,(SP)      ;MASK PROGRAMABLE
6355  031714  022726  040700          CMP     #ERRIVVIDPRIDRY,(SP)+
6356                                                          ;RHDS1 SHOULD HAVE THESE BITS
6357                                                          ;BUT ATA SHOULD BE CLEARED
6358  031720  001401                  BEQ     13$             ;CHECK RHER1 IF GOOD
6359  031722  104020                  ERROR   20              ;MOVING "1" INTO RHAS AT THE
6360                                                          ;DRIVE BIT POSITION DID NOT
6361                                                          ;CLEAR ATA IN RHDS1
6362
6363  031724  022737  177777  001716  13$:    CMP     #-1,@#ER1       ;RHER1 SHOULD NOT CHANGE
6364                                                          ;BY CLEARING RHAS
6365  031732  001401                  BEQ     TST72           ;BRANCH IF GOOD
6366  031734  104020                  ERROR   20              ;RHER1 WAS CHANGED BY CLEARING
6367                                                          ;RHAS BY MOVING "1" INTO
6368                                                          ;THE DRIVE BIT POSITION
6369
6370
6371
6372
6373
```

```
                                ;;*******************************************************************
                                ;*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)
                                ;
                                ;;*******************************************************************
031736  000004                  TST72:  SCOPE
031740  012706  001000                  MOV     #STACK,SP       ;RESET STACK
                                ;
031744  012737  000072  002032          MOV     #TTWO,@#TSTNM   ;THIS SAVES TEST NUMBER
031752  004037  042452                  JSR     R0,@#CLAREA     ;CLEAR SIMULATED DISK
031756  051116                          .WORD   DISK            ;FROM
031760  052114                          .WORD   DISK+776        ;TO
031762  177400                          .WORD   177400          ;DATA
                                ;
031764  004037  042452                  JSR     R0,@#CLAREA     ;CLEAR READ INTO BUFFER
031770  003154                          .WORD   REINTO          ;FROM
031772  004152                          .WORD   REINTO+776      ;TO
031774  000000                          .WORD   0               ;DATA
                                ;
                                ;*THESE ARE TO SETUP FOR DISKLESS USE ONLY
                                ;
031776  012737  010000  047200          MOV     #FMT22,@#CYL    ;CYLINDER 0 16 BITS PER WORD FORMAT
032004  105037  047203                  CLRB    @#SECTR+1       ;TRACK 0
032010  112737  000001  047202          MOVB    #1,@#SECTR      ;SECTOR 1
032016  005037  047204                  CLR     @#KEY1          ;KEY1=0
032022  005037  047206                  CLR     @#KEY2          ;KEY2=0
032026  012737  000012  047260          MOV     #10.,@#DAWORD   ;NO. OF DATA WORDS
032034  005037  047210                  CLR     @#X             ;THIS IS A READ COMMAND
032040  004537  043742                  JSR     R5,@#CRC        ;GO TO CALCULATE CRC
032044  047200                          CYL
032046  051100                          WCRC
                                ;
                                ;*THESE ARE REGULAR SETUPS
                                ;
032050  004737  042534                  JSR     PC,@#CLDISK     ;SETUP GENERAL REGISTERS
032054  013711  002066                  MOV     @#READAT,@R1    ;READ DATA INTO RHCS1=70
032060  012777  177766  147544          MOV     #-10.,@RHWC     ;10 DATA WORDS
032066  012777  003154  147540          MOV     #REINTO,@RHBA   ;STARTING ADDRESS OF READ BUFFER
032074  112746  000001                  MOVB    #1,-(SP)        ;IN LOWER BYTE GET SECTOR 1
032100  112766  000000  000001          MOVB    #0,1(SP)        ;GET TRACK0 IN UPPER BYTE
032106  012677  147532                  MOV     (SP)+,@RHDST    ;TRACK/SECTOR IN RHDST
032112  012777  014000  147530          MOV     #FMT22!ECI,@RHOF ;16 BITS PER WORD
                                                                ;ECC CORRECTION INHIBIT BECAUSE
                                                                ;ECC IS NOT CHECKED HERE
032120  005077  147526                  CLR     @RHCA           ;CYLINDER 0
032124  004737  042570                  JSR     PC,@#CHECKT     ;CHECK THAT DVA,RDY,DPR,DRY = 1
032130  104401  062145                  TYPE    ,CPHALT         ;AND THAT NO OTHERS = 1.  CANNOT CON-
                                                                ;TINUE TESTING IF BOTH AREN'T TRUE
032134  000000                          HALT                    ;STOP THE TEST
032136  052777  000010  147472          BIS     #BAI,@RHCS2     ;SET BUS ADDRESS INHIBIT
```

```
6430  032144  005037  002006                  CLR     @#ERFLG$        ;CLEAR ERROR FLAG
6431  032150  004737  047040                  JSR     PC,@#COMHD      ;READ DATA
6432
6433                                          ;*IF THE PROGRAM COMES BACK HERE WITHOUT ERROR PRINTOUT
6434                                          ;*FROM "COMHD" ROUTINE IN MEANS DATA IS TO BE CHECKED
6435
6436                                          ;*NOW THE DATA READ INTO "REINTO" BUFFER WILL
6437                                          ;*BE CHECKED, ONLY ONE WORD SHOULD BE CHANGED
6438                                          ;*ALL OTHER WORDS SHOULD REMAIN UNCHANGED
6439                                          ;*THE "WRFROM" BUFFER IS FILLED WITH EXPECTED DATA AND CHECKED
6440
6441  032154  005037  002006                  CLR     @#ERFLG$        ;CLEAR FLAG
6442  032160  004037  042452                  JSR     R0,@#CLAREA     ;CLEAR BUFFER
6443  032164  002110                          WRFROM                  ;FROM
6444  032166  003106                          WRFROM+776              ;TO
6445  032170  000000                          0                       ;DATA
6446
6447                                          ;*EXPECTED DATA IS 177400 IN FIRST LOCATION ONLY
6448  032172  012737  177400  002110          MOV     #177400,@#WRFROM        ;EXPECTED DATA
6449
6450                                          ;*NOW READ DATA BUFFER IS CHECKED
6451
6452  032200  012700  002110                  MOV     #WRFROM,R0      ;GOOD DATA
6453  032204  012701  003154                  MOV     #REINTO,R1      ;DATA READ
6454  032210  012702  000400                  MOV     #256.,R2        ;COUNTER
6455  032214  012737  000401  047320  1$:     MOV     #257.,@#ERWORD  ;FOR ERROR WORD NO
6456  032222  022021                          CMP     (R0)+,(R1)+     ;COMPARE GOOD WITH READ BUFFER
6457  032224  001424                          BEQ     2$              ;BRANCH IF GOOD
6458  032226  014037  001124                  MOV     -(R0),@#$GDDAT  ;GOOD DATA
6459  032232  014137  001126                  MOV     -(R1),@#$BDDAT  ;BAD DATA
6460  032236  160237  047320                  SUB     R2,@#ERWORD     ;ERROR WORD NO
6461  032242  005737  002006                  TST     @#ERFLG$
                                                        ;ANY ERRORS ALREADY THERE
6462  032246  001002                          BNE     3$              ;IF YES BRANCH DO NOT TYPE HEADER
6463  032250  104004                          ERROR   4               ;ERROR ON READ DATA
6464  032252  000401                          BR      4$              ;BRANCH TO AVOID PRINTING NEXT ERROR
6465  032254  104005          3$:             ERROR   5               ;WORD NO 1-10 ARE DATA
6466                                                                  ;WORDS
6467                                                                  ;WORD NOS 11-256 HAVE NOT BEEN
6468                                                                  ;READ AND BUFFER SHOULD BE
6469                                                                  ;ZERO IF OTHER THAN ZERO
6470                                                                  ;WRONG NUMBER OF WORDS HAVE
6471                                                                  ;BEEN READ IN THE DISK NOW
6472                                                                  ;CONTAINS 177400 ALL 256
6473                                                                  ;WORDS BUT ONLY 10 WORDS
6474                                                                  ;SHOULD BE READ IN
6475
6476  032256  022021          4$:             CMP     (R0)+,(R1)+     ;UNDO -(R0) AND -(R1) FOR ERROR
6477  032260  017746  146654                  MOV     @SWR,-(SP)      ;GET SWITCH SETTING
6478  032264  042716  177177                  BIC     #177177,(SP)    ;KEEP ONLY SWITCH 7 AND 8
6479  032270  022726  000200                  CMP     #SW07,(SP)+     ;IS 7 SET AND 8 RESET
6480  032274  001402                          BEQ     TST73           ;BRANCH OUT IF YES
6481  032276  005302          2$:             DEC     R2              ;COUNT
6482  032300  001345                          BNE     1$              ;BRANCH IF NOT COMPLETE
6483
6484
6485
```

```
 6486
 6487
```

```
6488
6489                                ;;*******************************************************************
6490                                ;*TEST 73       RHCS2 - BIT # 11 - NEM
6491
6492                                ;**     READ CYLINDER0, FORMAT 16 BITS PER WORD
6493                                ;**     TRACK0, SECTOR 1, KEYS 0, 1 WORD OF 177400
6494                                ;**     THIS IS DONE WITH BUS ADDRESS INHIBIT SET
6495                                ;**     BUS ADDRESS USED IS 760000 THIS IS ALWAYS NON EXISTANT
6496                                ;**     THIS SHOULD SET NEM
6497
6498                                ;;*******************************************************************
6499  032302  000004                TST73:  SCOPE
6500  032304  012706  001000                MOV     #STACK,SP       ;RESET STACK
6501  032310  012737  000073  002032        MOV     #TTNO,@#TSTNM   ;THIS SAVES TEST NUMBER
6502
6503                                ;*CHECK TO SEE IF THE PROGRAM IS RUNNING WITH AN RH70
6504
6505  032316  005737  002040                TST     @#RH70          ;TEST FLAG FOR RH70 CONTROLLER
6506  032322  001402                        BEQ     30$             ;IF FLAG = 1, THIS TEST IS SKIPPED
6507  032324  000137  032646                JMP     TST74           ;JUMP TO NEXT TEST ------------------)
6508  032330                        30$:                            ;IF FLAG = 0, DO THIS TEST
6509
6510
6511  032330  004037  042452                JSR     R0,@#CLAREA     ;CLEAR SIMULATED DISK
6512  032334  051116                        .WORD   DISK            ;FROM
6513  032336  052114                        .WORD   DISK+776        ;TO
6514  032340  177400                        .WORD   177400          ;DATA
6515
6516
6517                                ;*THESE ARE TO SETUP FOR DISKLESS USE ONLY
6518
6519  032342  012737  010000  047200        MOV     #FMT22,@#CYL    ;CYLINDER 0, 16 BITS PER WORD FORMAT
6520  032350  105037  047203              CLRB    @#SECOTR+1      ;TRACK 0
6521  032354  112737  000001  047202      MOVB    #1,@#SECOTR     ;SECTOR 1
6522  032362  005037  047204              CLR     @#KEY1          ;KEY1=0
6523  032366  005037  047206              CLR     @#KEY2          ;KEY2=0
6524  032372  012737  000001  047260      MOV     #1.,@#DAWORD    ;NO. OF DATA WORDS
6525  032400  005037  047210              CLR     @#X             ;THIS IS A READ COMMAND
6526  032404  004537  043742              JSR     R5,@#CRC        ;GO TO CALCULATE CRC
6527  032410  047200                      CYL
6528  032412  051100                      WCRC
6529
6530                                ;*THESE ARE REGULAR SETUPS
6531
6532  032414  004737  042534              JSR     PC,@#CLDISK     ;SETUP GENERAL REGISTERS
6533  032420  013711  002066              MOV     @#READAT,@R1    ;READ DATA INTO RHCS1=70
6534  032424  012777  177777  147200      MOV     #-1.,@RHWC      ;10 DATA WORDS
6535  032432  012777  160000  147174      MOV     #160000,@RHBA   ;STARTING ADDRESS OF READ BUFFER
6536  032440  052711  001400              BIS     #A16!A17,@R1    ;IS 760000
6537  032444  112746  000001              MOVB    #1,-(SP)        ;IN LOWER BYTE GET SECTOR 1
6538  032450  112766  000000  000001      MOVB    #0,1(SP)        ;GET TRACK0 IN UPPER BYTE
6539  032456  012677  147162              MOV     (SP)+,@RHDST    ;TRACK/SECTOR IN RHDST
6540  032462  012777  014000  147160      MOV     #FMT22!ECI,@RHOF ;16 BITS PER WORD
6541                                                              ;ECC CORRECTION INHIBIT BECAUSE
6542                                                              ;ECC IS NOT CHECKED HERE
6543  032470  005077  147156              CLR     @RHCA           ;CYLINDER 0
```

```
6544  032474  004737  042570                  JSR     PC,@#CHECKT     ;CHECK THAT DVA,RDY,DPR,DRY = 1
6545  032500  104401  062145                  TYPE    ,CPHALT         ;AND THAT NO OTHERS = 1.  CANNOT CON-
6546                                                                  ;TINUE TESTING IF BOTH AREN'T TRUE
6547  032504  000000                          HALT                    ;STOP THE TEST
6548  032506  052777  000010  147122          BIS     #PAI,@RHCS2     ;SET BUS ADDRESS INHIBIT
6549  032514  005037  002006                  CLR     @#ERFLG$        ;CLEAR ERROR FLAG
6550  032520  004737  047040                  JSR     PC,@#COMHD      ;READ DATA
6551
6552
6553
6554  032524  011137  001126          1$:     MOV     @R1,@#$BDDAT    ;TEST DATA
6555
6556  032530  022737  145670  001126          CMP     #SC!TRE!DVA!A16!A17!RDY!70,@#$BDDAT ;COMPARE RHCS1
6557  032536  001406                          BEQ     2$              ;BRANCH IF GOOD
6558  032540  012737  144270  001124          MOV     #SC!TRE!DVA!RDY!70,@#$GDDAT ;GOOD DATA
6559  032546  010137  042204                  MOV     R1,@#REGADR     ;REGISTER RHCS1
6560  032552  104001                          ERROR   1               ;REFERENCE NON EXISTANT
6561                                                                  ;MEMORY DID NOT SET
6562                                                                  ;REQUIRED BITS
6563  032554  013746  001774          2$:     MOV     @#UNIT,-(SP)    ;GET UNIT NUMBER
6564  032560  052716  004110                  BIS     #NEM!IR!BAI,(SP)        ;INCLUDE NEM BAI AND IR
6565  032564  012637  001124                  MOV     (SP)+,@#$GDDAT  ;
6566  032570  011237  001126                  MOV     @R2,@#$BDDAT    ;TEST DATA
6567  032574  023737  001124  001126          CMP     @#$GDDAT,@#$BDDAT;COMPARE RHCS2
6568  032602  001403                          BEQ     3$
6569  032604  010237  042204                  MOV     R2,@#REGADR     ;REGISTER ADDRESS
6570  032610  104001                          ERROR   1               ;REFRENCING NONEXISTANT MEMORY
6571                                                                  ;CAUSED AN ERROR SHOULD SET NEM
6572  032612  017737  147016  001126  3$:     MOV     @RHBA,@#$BDDAT  ;TEST DATA
6573
6574  032620  022737  160000  001126          CMP     #160000,@#$BDDAT        ;COMPARE RHBA
6575  032626  001407                          BEQ     4$              ;BRANCH IF GOOD
6576  032630  012737  160000  001124          MOV     #160000,@#$GDDAT;GOOD DATA
6577  032636  013737  001634  042204          MOV     @#RHBA,@#REGADR ;REGISTER ADDRESS RHBA
6578  032644  104001                          ERROR   1               ;AFTER A NON EXISTANT MEMORY ERROR
6579                                                                  ;RHBA DOES NOT HAVE 160002
6580  032646                          4$:
6581
6582
6583
6584
6585
6586
6587
```

```
6588
6589                                    ;;*******************************************************************
6590                                    ;*TEST 74       WRITE CHECK ERROR
6591
6592                                    ;**     WRITE CHECK DATA CYLINDER 0 FORMAT 16 BITS PER WORD
6593                                    ;**     TRACK 1, SECTOR 1, KEYS 0, 32 WORDS OF DATA
6594                                    ;**     FIFTH WORD IS CHANGED ON DISK TO GIVE WRITE CHECK ERROR
6595                                    ;**     ANY DEVICE LOGIC ERROR INDICATIONS ARE NOT CONCLUSIVE
6596                                    ;**     ON FIRST PASS
6597                                    ;**     BECAUSE ERROR LOGIC HAS NOT YET BEEN CHECKED
6598                                    ;**     ONLY RH WRITE CHECK ERROR IS TESTED
6599
6600                                    ;;*******************************************************************
6601  032646  000004                    TST74:  SCOPE
6602
6603                                            ;*DATA TABLE
6604                                            ;*TOTAL OF 32 WORDS CONSISTING OF
6605                                            ;*16 WORDS OF FLOATING ONES (EG. 1, 2, 4, 10)
6606                                            ;*16 WORDS OF FLOATING ZEROS (EG. 177776, 177775)
6607
6608
6609  032650  012706  001000                    MOV     #STACK,SP       ;RESET STACK
6610  032654  012737  000074  002032            MOV     #TTWO,@#TSTNM   ;THIS SAVES TEST NUMBER
6611  032662  004737  042534                    JSR     PC,@#CLDISK     ;INIT AND SET UP GENERAL REGISTERS
6612
6613                                            ;*SET UP "REINTO" FOR WHAT IS TO BE READ
6614
6615  032666  012700  000001                    MOV     #1,R0           ;GETTING READY TO FLOAT 1
6616  032672  012701  003154                    MOV     #REINTO,R1      ;STARTING ADDRESS WHERE 1 GOES
6617  032676  010021                    1$:     MOV     R0,(R1)+        ;MOVE FLOATING 1
6618  032700  006100                            ROL     R0              ;GET 1 ONE BIT LEFT
6619  032702  103375                            BCC     1$              ;BRANCH IF 16 NOT DONE
6620  032704  012700  177776                    MOV     #177776,R0      ;GETTING READY TO FLOAT 0
6621  032710  012701  003214                    MOV     #REINTO+<16.*2>,R1 ;STARTING ADDRESS WHERE 177776 GOES
6622  032714  010021                    2$:     MOV     R0,(R1)+        ;MOVE IN FLOATING 0
6623  032716  000261                            SEC                     ;SET CARRY
6624  032720  006100                            ROL     R0              ;GET 0 ONE BIT LEFT
6625  032722  103774                            BCS     2$              ;BRANCH IF 16 NOT DONE
6626
6627  032724  004037  042452                    JSR     R0,@#CLAREA     ;FILL REST OF BUFFER WITH 1
6628  032730  003254                            .WORD   REINTO+<32.*2>  ;FROM
6629  032732  004152                            .WORD   REINTO+776      ;TO
6630  032734  000001                            .WORD   1               ;WITH DATA
6631
6632                                            ;*SET UP SIMULATED DISK WITH WHAT IS TO BE READ
6633
6634  032736  012700  000001                    MOV     #1,R0           ;GETTING READY TO FLOAT 1
6635  032742  012701  051116                    MOV     #DISK,R1        ;STARTING ADDRESS WHERE 1 GOES
6636  032746  010021                    3$:     MOV     R0,(R1)+        ;MOVE FLOATING 1
6637  032750  006100                            ROL     R0              ;GET 1 ONE BIT LEFT
6638  032752  103375                            BCC     3$              ;BRANCH IF 16 NOT DONE
6639
6640  032754  012700  177776                    MOV     #177776,R0      ;GETTING READY TO FLOAT 0
6641  032760  012701  051156                    MOV     #DISK+<16.*2>,R1 ;STARTING ADDRESS WHERE 177776 GOES
6642  032764  010021                    4$:     MOV     R0,(R1)+        ;MOVE FLOATING 0
6643  032766  000261                            SEC                     ;SET CARRY
```

```
6644  032770  006100                          ROL     R0              ;GET 0 ONE BIT LEFT
6645  032772  103774                          BCS     4$              ;BRANCH IF 16 NOT DONE
6646
6647  032774  004037  042452                  JSR     R0,@#CLAREA     ;FILL REST OF BUFFER WITH 0
6648  033000  051216                          .WORD   DISK+<32.*2>    ;FROM
6649  033002  052114                          .WORD   DISK+776        ;TO
6650  033004  000000                          .WORD   0               ;WITH DATA
6651
6652                                          ;*CHANGE FIFTH WORD TO 0 ON DISK
6653
6654  033006  005037  051126                  CLR     @#DISK+10       ;CLEAR FIFTH WORD ON DISK
6655  033012  005037  002006                  CLR     @#ERFLG$        ;CLEAR ERROR FLAG
6656  033016  004737  043572                  JSR     PC,@#WRCHDA     ;WRITE CHECK DATA
6657                                                                  ;CYLINDER 0, TRACK 1, SECTOR 1
6658                                                                  ;KEYS 0, 32 WORDS.
6659
6660                                          ;*IF THE PROGRAM COMES BACK HERE THEN WRITE CHECK
6661                                          ;*HAS BEEN COMPLETED, NOW WRITE CHECK ERROR BIT IS TESTED
6662                                          ;*ALONG WITH RHWC FOR PROPER WORD COUNT AND RHBA FOR ADDRESS
6663
6664  033022  013746  001774                  MOV     @#UNIT,-(SP)    ;GET UNIT NUMBER
6665  033026  052716  040300                  BIS     #IR!DR!WCE,(SP) ;ONLY BIT 6 SHOULD BE SET
6666  033032  004737  042140                  JSR     PC,@#PUTREG     ;SAVE REGISTERS
6667  033036  022637  001712                  CMP     (SP)+,@#CS2     ;COMPARE RHCS2
6668  033042  001407                          BEQ     6$              ;BRANCH IF GOOD
6669  033044  032737  040000  001712          BIT     #WCE,@#CS2      ;WRITE CHECK ERROR HIGH?
6670  033052  001002                          BNE     5$              ;BRANCH IF ERROR NOT DUE TO "WCE"
6671  033054  104017                          ERROR   17              ;RHDB CONTAINS FAILING WORD
6672  033056  000401                          BR      6$              ;RHBA CONTAINS ADDRESS+2
6673                                                                  ;OF THE WORD IN MEMORY FROM
6674                                                                  ;THE DISK THAT DID NOT COMPARE
6675                                                                  ;TRE AND SC WILL BE SET DUE TO WCE
6676  033060  104017                  5$:     ERROR   17              ;WCE WAS CORRECTLY NOT SET
6677                                                                  ;BUT SOME BITS OTHER THAN
6678                                                                  ;IR AND UNIT NO. WERE SET
6679
6680  033062  005737  002040          6$:     TST     @#RH70          ;TEST FOR RH70 CONTROLLER
6681  033066  001414                          BEQ     16$             ;SKIP RH70 CODE AND DO RH11 IF NOT
6682
6683  033070  022737  177750  001706          CMP     #-24.,@#WC      ;COMPARE RHWC AFTER A FORCED
6684                                                                  ;WRITE CHECK ERROR
6685  033076  001402                          BEQ     17$             ;CHECK RHBA IF GOOD
6686  033100  104017                          ERROR   17              ;WORD COUNT REGISTER IN ERROR AFTER A
6687                                                                  ;FORCED WRITE CHECK ERROR ON FIFTH WORD
6688  033102  000421                          BR      15$             ;BRANCH TO CONTINUE TEST
6689
6690  033104  022737  003174  001710  17$:     CMP     #REINTO+<8.*2>,@#BA ;COMPARE RHBA AFTER A FORCED
6691                                                                  ;WRITE CHECK ERROR IN FIFTH WORD
6692  033112  001415                          BEQ     15$             ;CONTINUE IF GOOD
6693  033114  104017                          ERROR   17              ;BUS ADDRESS REGISTER IN ERROR AFTER
6694                                                                  ;FORCED WRITE CHECK ERROR ON FIFTH WORD
6695  033116  000413                          BR      15$             ;SKIP RH11 CODE AND CONTINUE WITH TEST
6696
6697  033120  022737  177745  001706  16$:     CMP     #-27.,@#WC      ;COMPARE RHWC AFTER A FORCED
6698                                                                  ;WRITE CHECK ERROR
6699  033126  001402                          BEQ     14$             ;CHECK RHBA IF GOOD
```

H13

```
6700  033130  104017                          EPROR   17              ;WORD COUNT REGISTER IN ERROR AFTER A
6701                                                                  ;FORCED WRITE CHECK ERROR ON FIFTH WORD
6702  033132  000405                          BR      15$             ;BRANCH TO CONTINUE TEST
6703
6704  033134  022737  003166  001710  14$:    CMP     #REINTO+<5.*2>,@#BA;COMPARE RHBA AFTER FORCED
6705                                                                  ;WRITE CHECK ERROR IN FIFTH WORD
6706  033142  001401                          BEQ     15$             ;CONTINUE IF GOOD
6707  033144  104017                          ERROR   17              ;BUS ADDRESS REGISTER IN ERROR AFTER
6708                                                                  ;FORCED WRITE CHECK ERROR ON FIFTH WORD
6709
6710                                          ;*NOW CHECK MEMORY TO SEE IF ANYTHING GOT DESTROYED
6711                                          ;*FILL "WRFROM" WITH WHAT SHOULD BE IN REINTO THEN CHECK
6712
6713  033146  005037  002006          15$:    CLR     @#ERFLG$        ;CLEAR ERROR FLAG
6714  033152  012700  000001                  MOV     #1,R0           ;GETTING READY TO FLOAT 1
6715  033156  012701  002110                  MOV     #WRFROM,R1      ;START ADDRESS WHERE 1 GOES
6716  033162  010021                  7$:     MOV     R0,(R1)+        ;MOVE FLOATING 1
6717  033164  006100                          ROL     R0              ;GET 1 ONE BIT LEFT
6718  033166  103375                          BCC     7$              ;BRANCH IF 16 NOT DONE
6719
6720  033170  012700  177776                  MOV     #177776,R0      ;GETTING READY TO FLOAT 0
6721  033174  012701  002150                  MOV     #WRFROM+<16.*2>,R1 ;STARTING ADDRESS WHERE 177776 GOES
6722  033200  010021                  10$:    MOV     R0,(R1)+        ;MOVE IN FLOATING 0
6723  033202  000261                          SEC                     ;SET CARRY
6724  033204  006100                          ROL     R0              ;GET 0 ONE BIT LEFT
6725  033206  103774                          BCS     10$             ;BRANCH IF CARRY SET
6726
6727  033210  004037  042452                  JSR     R0,@#CLAREA     ;FILL REST OF BUFFER WITH 1
6728  033214  002210                          .WORD   WRFROM+<32.*2>  ;FROM
6729  033216  003106                          .WORD   WRFROM+776      ;TO
6730  033220  000001                          .WORD   1               ;WITH DATA
6731
6732                                          ;*NOW THE READ BUFFER WILL BE CHECKED
6733
6734  033222  004037  043430                  JSR     R0,@#COMPAR     ;CHECK
6735  033226  002110                          WRFROM                  ;GOOD BUFFER
6736  033230  003154                          REINTO                  ;TEST BUFFER
6737  033232  000400                          256.                    ;NUMBER OF WORDS CHECKED
6738  033234  033242                          11$                     ;PETURN POINT FOR ERROR HEADER
6739  033236  033246                          12$                     ;PETURN POINT FOR ERROR DATA
6740
6741  033240  033254                          TST75                   ;RETURN FOR GOOD COMPARISON
6742
6743  033242  104004                  11$:    ERROR   4               ;READ NEXT ERROR 5
6744  033244  000207                          RTS     PC              ;RETURN TO COMPARISON SUBROUTINE
6745  033246  104005                  12$:    ERROR   5               ;DATA IN REINTO BUFFER GOT
6746                                                                  ;CHANGED AFTER A WRITE
6747                                                                  ;CHECK DATA COMMAND
6748                                                                  ;WORD NO CONTAINS THE WORD
6749                                                                  ;NUMBER THAT GOT CHANGED
6750  033250  000207                          RTS     PC              ;RETURN TO COMPARISON SUBROUTINE
6751
6752  033252  000240                  13$:    NOP                     ;ONLY A BRANCH POINT
6753
6754
6755
```

```
6756
6757                                    ;;*********************************************************************
6758                                    ;*TEST 75       ERROR REGISTER #1-BIT 4 -FORMAT ERROR
6759                                    ;**     THE SIMULATED DISK IS FILLED WITH CYLINGER 0 TRACK 1
6760                                    ;**     SECTOR 0 FORMAT=16 BITS PER WORD AND 4 WORDS
6761                                    ;**     OF 125252, A READ HEADER AND DATA COMMAND IS GIVEN WITH 16 BITS
6762                                    ;**     PER WORD FORMAT, FER=BIT4 SHOULD SET BUT THE
6763                                    ;**     READ SHOULD BE COMPLETE
6764
6765                                    ;;*********************************************************************
6766  033254  000004                   TST75:  SCOPE
6767
6768
6769  033256  012706  001000                    MOV     #STACK,SP       ;RESET STACK
6770
6771  033262  012737  000075  002032            MOV     #TTNO,@#TSTNM   ;THIS SAVES TEST NUMBER
6772
6773                                    ;*      SETUP FOR WHAT IS TO BE READ
6774                                    ;*      HEADER CRC IS RESTORED FROM A SUBROUTINE
6775
6776  033270  012746  125252                    MOV     #125252,-(SP)   ;DATA TO BE READ
6777  033274  012705  000400                    MOV     #256.,R5        ;COUNTER
6778  033300  012700  051116                    MOV     #DISK,R0        ;START OF SIMULATED DISK DATA
6779  033304  011620                   1$:     MOV     (SP),(R0)+      ;MOVE IN DATA ON TO SIMULATED DISK
6780  033306  005305                            DEC     R5              ;COUNT
6781  033310  001375                            BNE     1$              ;BRANCH IF 256 NOT COMPLETE
6782  033312  005726                            TST     (SP)+           ;UNDO -(SP)
6783  033314  012705  000021                    MOV     #17.,R5         ;2 ECC WORDS
6784                                                                    ;1 DATA GAP
6785                                                                    ;14 TOLERANCE GAP
6786  033320  005020                   2$:     CLR     (R0)+           ;CLEAR ECC, DATA GAP, AND
6787  033322  005305                            DEC     R5              ;TOLERANCE GAP
6788  033324  001375                            BNE     2$              ;BRANCH IF NOT COMPLETE
6789
6790
6791                                    ;*THESE ARE TO SETUP FOR DISKLESS USE ONLY
6792
6793  033326  012737  000000  047200            MOV     #010,@#CYL ;16 BITS PER WORD
6794                                                                    ;CYLINDER 0, FORMAT 16 BITS
6795  033334  112737  000001  047203            MOVB    #1,@#SECOTR+1   ;TRACK 1
6796  033342  112737  000000  047202            MOVB    #0,@#SECOTR     ;SECTOR 0
6797  033350  012737  000000  047204            MOV     #0,@#KEY1       ;KEY1=0
6798  033356  012737  000000  047206            MOV     #0,@#KEY2       ;KEY2=0
6799  033364  012737  000004  047260            MOV     #4.,@#DAWORD    ;NO. OF DATA WORDS
6800  033372  005037  047210                    CLR     @#X             ;THIS IS A READ COMMAND
6801  033376  004537  043742                    JSR     R5,@#CRC        ;GO TO CALCULATE CRC
6802  033402  047200                            CYL
6803  033404  051100                            WCRC
6804
6805                                    ;*THESE ARE REGULAR SETUPS FOR RH11 AND "REINTO"
6806
6807  033406  004737  042534                    JSR     PC,@#CLDISK     ;SETUP GENERAL REGISTERS
6808  033412  012777  177770  146212            MOV     #-4.-4.,@RHWC   ;4. DATA 4 HEADER WORDS
6809  033420  012777  003154  146206            MOV     #REINTO,@RHBA   ;STARTING ADDRESS OF READ BUFFER
6810  033426  112746  000000                    MOVB    #0,-(SP)        ;IN LOWER BYTE GET SECTOR
6811  033432  112766  000001  000001            MOVB    #1,1(SP)        ;GET TRACK IN HIGHER BYTE
```

J13

```
6812  033440  012677  146200                  MOV     (SP)+,@RHDST    ;TRACK/SECTOR IN RHDST
6813  033444  012777  014000  146176          MOV     #FMT22!ECI,@RHOF ;16 BITS PER WORD
6814                                                                    ;ECC CORRECTION INHIBIT
6815                                                                    ;BECAUSE ECC IS NOT GOING
6816                                                                    ;TO BE CHECKED
6817  033452  005077  146174                  CLR     @RHCA           ;CYLINDER 0
6818
6819  033456  004737  042570                  JSR     PC,@#CHECKT     ;CHECK THAT DVA,RDY,DPR,DRY = 1
6820  033462  104401  062145                  TYPE    ,CPHALT         ;AND THAT NO OTHERS = 1.  CANNOT CON-
6821                                                                    ;TINUE TESTING IF BOTH AREN'T TRUE
6822  033466  000000                          HALT                    ;STOP THE TEST
6823
6824  033470  013711  002070                  MOV     @#REFOR,@R1     ;READ HEADER AND DATA=72
6825  033474  005037  002006                  CLR     @#ERFLG$        ;CLEAR ERROR FLAG
6826  033500  004737  047040                  JSR     PC,@#COMHD      ;READ HEADER AND DATA
6827
6828
6829                                          ;*IF THE PROGRAM COMES BACK HERE WITHOUT ERROR PRINTOUTS
6830                                          ;*FROM THE "COMHD" ROUTINE THAT MEANS SECTOR GAP,
6831                                          ;*FIRST SYNC, HEADER, HEADER CRC, HEADER GAP AND
6832                                          ;*SYNC BYTE HAVE GONE BY AND SYNCS WERE CORRECTLY
6833                                          ;*DETECTED.
6834
6835
6836                                          ;*RHWC IS CHECKED TO BE = 0 AFTER THE READ OPERATION
6837
6838  033504  017737  146122  001126          MOV     @RHWC,$BDDAT    ;LOAD AND TEST RHWC
6839  033512  001401                          BEQ     20$             ;SHOULD = 0
6840  033514  104040                          ERROR   40              ;RHWC DOES NOT = 0 AFTER A READ
6841                                                                  ;HEADER AND DATA
6842
6843                                          ;*HEADER AND DATA ARE TO BE CHECKED.
6844                                          ;*IN CHECKING READ DATA THE WRITE FROM BUFFER
6845                                          ;*"WRFROM" IS FILLED WITH EXPECTED DATA AND
6846                                          ;*COMPARISONS ARE MADE.
6847
6848  033516  005737  002006          20$:    TST     @#ERFLG$        ;ANY ERRORS ALREADY THERE
6849  033522  001055                          BNE     TST76           ;BRANCH IF YES
6850  033524  004737  042760                  JSR     PC,@#CHECKE     ;CHECK THAT BITS = 1
6851  033530  104401  062145                  TYPE    ,CPHALT         ;CANNOT CONTINUE TESTING IF THEY DON'T
6852  033534  000000                          HALT                    ;STOP THE TEST
6853  033536  012700  002110                  MOV     #WRFROM,R0      ;GETTING READY TO FILL EXPECTED DATA
6854  033542  012720  000000                  MOV     #0,(R0)+        ;CYLINDER 0
6855  033546  112746  000000                  MOVB    #0,-(SP)        ;IN LOWER BYTE GET SECTOR
6856  033552  112766  000001  000001          MOVB    #1,1(SP)        ;GET TRACK IN HIGHER BYTE
6857  033560  012620                          MOV     (SP)+,(R0)+     ;GET TRACK/SECTOR IN BUFFER
6858  033562  012720  000000                  MOV     #0,(R0)+        ;KEY1 IN BUFFER
6859  033566  012720  000000                  MOV     #0,(R0)+        ;KEY2 IN BUFFER
6860  033572  012701  000400                  MOV     #256.,R1        ;DATA WORD COUNTER
6861  033576  012702  125252                  MOV     #125252,R2              ;DATA
6862  033602  010220                  3$:     MOV     R2,(R0)+        ;DATA INTO BUFFER
6863  033604  005301                          DEC     R1              ;COUNT
6864  033606  001375                          BNE     3$              ;BRANCH IF 256 NOT DONE
6865
6866                                          ;*NOW READ DATA BUFFER WILL BE CHECKED
6867
```

K13

```
6868  033610  004037  043430              JSR     R0,@#COMPAR     ;CHECK
6869  033614  002110                      WRFROM                  ;GOOD BUFFER
6870  033616  003154                      REINTO                  ;TEST BUFFER
6871  033620  000010                      4+4.                    ;NUMBER OF WORDS CHECKED
6872  033622  033630                      4$                      ;RETURN POINT FOR ERROR HEADER
6873  033624  033634                      5$                      ;RETURN POINT FOR ERROR DATA
6874  033626  033640                      6$                      ;RETURN FOR GOOD COMPARISON
6875  033630  104004              4$:     ERROR   4               ;READ NEXT ERROR
6876  033632  000207                      RTS     PC              ;RETURN TO "COMPAR"
6877  033634  104005              5$:     ERROR   5               ;WORD NOS 1 TO 4 ARE
6878                                                              ;HEADER WORDS
6879                                                              ;5 TO 260 ARE DATA WORDS
6880  033636  000207                      RTS     PC              ;RETURN TO "COMPAR"
6881
6882
6883                                      ;*NOW SEE THAT FORMAT ERROR BIT GOT SET
6884
6885  033640  004737  042140      6$:     JSR     PC,@#PUTREG     ;SAVE REGISTERS
6886
6887  033644  022737  100020  001716      CMP     #FERIDCK,@#ER1  ;FORMAT ERROR SHOULD BE SET
6888  033652  001401                      BEQ     TST76           ;BRANCH IF GOOD
6889  033654  104020                      ERROR   20              ;A 16 BIT PER WORD READ WAS ATTEMPTED
6890                                                              ;WHEN THE DISK HAD
6891                                                              ;THE FORMAT BIT=0= 18 BITS PER
6892                                                              ;WORD THE READ WAS
6893                                                              ;COMPLETED BUT ERROR REG
6894                                                              ;WAS NOT RIGHT
6895                                                              ;NOTE DCK WILL BE SET BECAUSE
6896                                                              ;ECC HAS NOT BEEN GENERATED
6897
6898
```

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

                                ;;*****************************************************************
033656  000004                TST76:  SCOPE
                                        ;*NOW A WRITE DATA WILL BE ATTEMPTED WITH
                                        ;*WRONG FORMAT BIT

033660  012706  001000                MOV     #STACK,SP       ;RESET STACK

033664  012737  000076  002032        MOV     #TTWO,@#TSTNM   ;THIS SAVES TEST NUMBER

033672  012737  177777  047314        MOV     #-1,@#NOSYNC    ;SET FLAG SO THAT DATA SYNC
                                                              ;AND DATA IS NOT READ
033700  004037  042452          FRMAT1: JSR     R0,@#CLAREA     ;CLEAR SIMULATED DISK
033704  051116                        .WORD   DISK            ;FROM
033706  052142                        .WORD   TOLGAP+16       ;TO
033710  000000                        .WORD   0               ;DATA
                                ;*THESE ARE SETUP FOR DISKLESS USE ONLY
033712  005037  047200                CLR     @#CYL           ;CYLINDER 0, FORMAT 18 BIT WORDS
033716  105037  047203                CLRB    @#SECTR+1       ;TRACK 0
033722  105037  047202                CLRB    @#SECTR         ;SECTOR 0
033726  005037  047204                CLR     @#KEY1          ;KEY1 0
033732  005037  047206                CLR     @#KEY2          ;KEY2 0
033736  012737  000004  047246        MOV     #4,@#NOWORD     ;NO OF DATA WORDS
033744  012737  000001  047210        MOV     #1,@#X          ;WRITE DATA
033752  004537  043742                JSR     R5,@#CRC        ;GO TO CALCULATE CRC
033756  052336                        WCYL
033760  052346                        GCRC

                                ;*THESE AER REGULAR SETUPS

033762  004037  042452                JSR     R0,@#CLAREA     ;FILL WRITE FROM BUFFER WITH 125252
033766  002110                        WRFROM                  ;FROM
033770  002116                        WRFROM+6                ;TO
033772  125252                        125252                  ;DATA
033774  004737  042534                JSR     PC,@#CLDISK     ;SETUP GENERAL REGISTERS
034000  012777  177774  145624        MOV     #-4,@RHWC       ;256 DATA WORDS
034006  012777  002110  145620        MOV     #WRFROM,@RHBA   ;STARTING ADDRESS OF WRITE BUFFER
034014  005077  145624                CLR     @RHDST          ;TRACK=0 SECTOR=0
034020  012777  010000  145622        MOV     #FMT22,@RHOF    ;16 BITS PER WORD FORMAT
034026  005077  145620                CLR     @RHCA           ;CYLINDER 0
034032  004737  042570                JSR     PC,@#CHECKT     ;CHECK THAT DVA,RDY,DPR,DRY = 1
034036  104401  062145                TYPE    ,CPHALT         ;AND THAT NO OTHERS = 1.  CANNOT CON-
                                                              ;TINUE TESTING IF BOTH AREN'T TRUE
034042  000000                        HALT                    ;STOP THE TEST
034044  013711  002062                MOV     @#WRIDAT,@R1    ;WRITE DATA=60
034050  005037  002006                CLR     @#ERFLG$        ;CLEAR ERROR FLAG
034054  004737  047040                JSR     PC,@#COMHD      ;WRITE DATA
                                ;*IF THE PROGRAM COMES BACK HERE WITHOUT ERROR PRINTOUTS
```

M13

```
                                ;*FROM THE "COMMD" ROUTINE IT MEANS SECTOR GAP, SYNC BYTE
                                ;*HEADER, HEADER CRC, HEADER GAP AND SYNC BYTE HAVE GONE BY
                                ;*AND SYNCS WERE CORRECTLY DETECTED
                                ;*DATA IS TO BE CHECKED
034060  004737  042140          JSR     PC,@#PUTREG     ;SAVE REGISTERS
034064  005737  002006          TST     @#ERFLG$        ;HAS ANY ERRORS OCCURED?
034070  001041                  BNE     4$              ;BRANCH IF YES
034072  012700  000000          MOV     #0,R0           ;GOOD DATA
034076  012701  051116          MOV     #DISK,R1        ;DATA WRITTEN INTO "DISK"
034102  012702  000004          MOV     #4,R2           ;COUNTER
034106  012737  000005  047320  1$:     MOV     #5,@#ERWORD     ;FOR ERROR WORD
034114  020021                  CMP     R0,(R1)+        ;COMPARE GOOD DATA WITH DATA ON DISK
034116  001424                  BEQ     3$              ;BRANCH IF GOOD
034120  010037  001124          MOV     R0,@#$GDDAT     ;GOOD DATA
034124  014137  001126          MOV     -(R1),@#$BDDAT  ;BAD DATA
034130  160237  047320          SUB     R2,@#ERWORD     ;ERROR WORD NO
034134  005737  002006          TST     @#ERFLG$        ;ANY ERRORS ALREADY THERE?
034140  001002                  BNE     2$              ;BRANCH IF YES
034142  104004                  ERROR   4               ;ERROR ON WRITE DATA COMMAND
                                                        ;ON A WRITE DATA WITH
                                                        ;WRONG FORMAT NO DATA
                                                        ;SHOULD BE WRITTEN
                                                        ;WORD NO GIVES WORD IN ERROR
034144  000401                  BR      5$              ;BRANCH TO AVOID PRINTING NEXT ERROR
034146  104005          2$:     ERROR   5
034150  005721          5$:     TST     (R1)+           ;UNDO -(R1) FOR BAD DATA
034152  017746  144762          MOV     @SWR,-(SP)      ;GET SWITCH SETTING
034156  042716  177177          BIC     #177177,(SP)    ;KEEP ONLY SWITCH 7 AND 8
034162  022726  000200          CMP     #SW07,(SP)+     ;IS 7 SET AND 8 RESET.
034166  001402                  BEQ     4$              ;BRANCH IF YES
034170  005302          3$:     DEC     R2              ;IF NOT COUNT 256 WORDS
034172  001345                  BNE     1$              ;BRANCH IF 256 NOT DONE

                                ;*NOW CHECK TO SEE THAT FORMAT ERROR BIT GOT SET

034174  022737  000020  001716  4$:     CMP     #FER,@#ER1      ;FORMAT ERROR SHOULD BE SET
034202  001401                  BEQ     TST77           ;BRANCH IF GOOD
034204  104020                  ERROR   20              ;A 16 BIT PER WORD WRITE DATA
                                                        ;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
                                ;
                                ;;*****************************************************************
034206  000004                  TST77:  SCOPE
                                ;
034210  012706  001000                  MOV     #STACK,SP       ;RESET STACK
034214  012737  000077  002032          MOV     #TTNO,@#TSTNM   ;THIS SAVES TEST NUMBER
034222  004737  042534                  JSR     PC,@#CLDISK     ;CLEAR DISK
034226  004737  042570                  JSR     PC,@#CHECKT     ;CHECK THAT DVA,RDY,DPR,DRY = 1
034232  104401  062145                  TYPE    ,CPHALT         ;AND THAT NO OTHERS = 1.  CANNOT CON-
                                                                ;TINUE TESTING IF BOTH AREN'T TRUE
034236  000000                          HALT                    ;STOP THE TEST
034240  012700  001652                  MOV     #RHCA,R0
034244  012005                  1$:     MOV     (R0)+,R5        ;R5 HAS ADDRESS OF REG. UNDER TEST
034246  052777  000040  145362  2$:     BIS     #CLR,@RHCS2
034254  013777  001774  145354          MOV     @#UNIT,@RHCS2   ;REINSTATE UNIT NO.
                                ;
                                ;*SET UP FOR AN OPERATION (WRITE HEADER AND DATA)
                                ;
034262  013777  002064  145350          MOV     @#WRIFOR,@RHCS1 ;WRITE HEADER AND DATA=62
                                                                ;IN RHCS1
034270  012777  177766  145334          MOV     #-10.,@RHWC     ;10 WORDS
034276  012777  002110  145330          MOV     #WRFROM,@RHBA   ;BUS ADDRESS = WRFROM
034304  012777  000010  145332          MOV     #10,@RHDST      ;DESIRED TRACK=0, SECTOR=10
034312  052777  000010  145316          BIS     #BAI,@RHCS2     ;BUS ADDRESS INCREMENT INHIBIT
034320  012777  010000  145322          MOV     #FMT22,@RHOF    ;FORMAT 16 BIT WORDS
034326  005077  145320                  CLR     @RHCA           ;CYLINDER =0
                                ;
                                ;*SAVE REGISTERS
                                ;
034332  004037  043226                  JSR     R0,@#SAVER      ;SAVE
034336  001640                  RHCS1                           ;FROM
034340  003154                  REINTO                          ;TO
034342  000016                  14.                             ;NUMBER OF REGISTERS SAVED
                                ;
                                ;*NOW THE COMMAND IS GIVES TO
                                ;*WRITE HEADER AND DATA FOR CYL=0, SECTOR=10
                                ;*TRACK=0 IT COMES BACK AFTER ONE SECTOR
                                ;*HAS PASSED
                                ;
```

```
7055  034344  012777  000001  145306          MOV     #DMD,@RHMR      ;SET DIAGNOSTIC MODE
7056  034352  005277  145262                  INC     @RHCS1          ;GO TO RHCS1 WITH 62
7057  034356  012715  177672                  MOV     #177672,@R5     ;TRY WRITING ALL BITS EXCEPT
7058                                                                  ;GO, RMR, IE
7059  034362  052737  000001  003174          BIS     #DMD,@#REINTO+20 ;SET DMD IN SAVED REGISTER RHMR
7060  034370  052737  000004  003156          BIS     #RMR,@#REINTO+2 ;SET RMR IN SAVED REG. RHER1
7061  034376  042737  000200  003176          BIC     #DRY,@#REINTO+22        ;CLEAR DRY IN RHDS1
7062  034404  052737  040000  003176          BIS     #ERR,@#REINTO+22        ;SET ERR IN RHDS1
7063  034412  052737  000001  003154          BIS     #GO,@#REINTO    ;SET GO IN SAVED REG. RHCS1
7064  034420  042737  000200  003154          BIC     #RDY,@#REINTO   ;CLEAR RDY BIT
7065
7066                                          ;*AFTER AN ATTEMPT TO WRITE INTO A REGISTER
7067                                          ;*SAVE REGISTERS AGAIN SO THAT COMPARES CAN BE DONE
7068
7069  034426  004037  043226                  JSR     R0,@#SAVER      ;SAVE
7070  034432  001640                          RHCS1                   ;FROM
7071  034434  002110                          WRFROM                  ;TO
7072  034436  000016                          14.                     ;NUMBER
7073
7074                                          ;*AS UPPER BYTE OF RHAS CAN BE CHANGING IN A DUAL PORT
7075                                          ;*OPERATION THE UPPER BYTE OF RHAS WILL BE SAVED AS IS
7076                                          ;*SO THAT THE COMPARES ARE ONLY VALID FOR THE LOWER BYTE
7077  034440  113737  003173  002127          MOVB    @#REINTO+17,@#WRFROM+17;SAVE UPPER RHAS
7078
7079
7080                                          ;*COMPARE REGISTERS BEFORE ATTEMPTED WRITE WITH AFTER
7081
7082  034446  004037  043430                  JSR     R0,@#COMPAR     ;COMPAR
7083  034452  003154                          REINTO                  ;GO BUFFER
7084  034454  002110                          WRFROM                  ;TEST BUFFER
7085  034456  000016                          14.                     ;NUMBER
7086  034460  034466                          4$                      ;RETURN FOR ERROR
7087  034462  034466                          4$                      ;SAME
7088  034464  034506                          5$                      ;RETURN FOR GOOD COMPARISON
7089  034466  013705  047320          4$:     MOV     @#ERWORD,R5     ;GETTING READY TO INDEX
7090  034472  060505                  ADD     R5,R5                   ;DOUBLE ERROR WORD
7091  034474  016537  001636  042204          MOV     RHCS1-2(R5),@#REGADR ;FAILING REG. ADDRESS
7092  034502  104001                  ERROR   1                       ;CONTENTS OF REGISTER
7093  034504  000207                  RTS     PC                      ;CHANGED WITH
7094                                                                  ;AN ATTEMPT TO WRITE
7095                                                                  ;DURING AN OPERATION
7096                                          ;*THE FOLLOWING CLEAR MAY SET THE ATA BIT BECAUSE GO IS HIGH
7097                                          ;*
7098  034506  004737  042534          5$:     JSR     PC,@#CLDISK     ;CLEAR DISK
7099
7100
```

```
7101
7102
7103                                    ;;*****************************************************************
7104                                    ;*TEST 100      MAKE CURRENT CYLINDER = 1
7105                                    ;;*****************************************************************
7106  034512  000004                    TST100: SCOPE
7107  034514  012706  001000                    MOV     #STACK,SP       ;RESET STACK
7108  034520  012737  000100  002032            MOV     #TTNO,@#TSTNM   ;THIS SAVES TEST NUMBER
7109  034526  004737  042534                    JSR     PC,@#CLDISK     ;INIT DRIVE
7110  034532  012777  000001  145120            MOV     #DMD,@RHMR      ;SET DIAGNOSTIC MODE
7111  034540  004037  045106                    JSR     R0,@#MAKECYL    ;SUBROUTINE TO GIVE A SEEK
7112                                                                    ;COMMAND FOLOWED BY AN INIT
7113                                                                    ;THIS SHOULD CHANGE RHCC
7114  034544  000001                            1                       ;CHANGE RHCC TO 1
7115
7116
7117
7118                                    ;;*****************************************************************
7119                                    ;*TEST 101      ERROR REG1 - BIT #7 - HEADER COMPARE ERROR
7120
7121                                    ;**     THE SIMULATED DISK IS SET TO READ CYLINDER=0, TRACK=1
7122                                    ;**     SECTOR=1, KEYS=1, 256 WORDS OF 177400
7123                                    ;**     A READ HEADER AND DATA COMMAND IS GIVEN TO READ
7124                                    ;**     CYLINDER=1, TRACK=1, SECTOR=1, KEY1=1, KEY2=1
7125                                    ;**     REINTO BUFFER IS FILLED WITH 0
7126                                    ;**     WRFROM IS FILLED WITH 10000,401,1,1,1, AND ALL 177400
7127                                    ;**     AFTER THE READ THE REINTO BUFFER IS EXPECTED TO
7128                                    ;**     HAVE WHAT IS IN WRFROM - 10000,401,1,1 AND ALL 177400
7129
7130                                    ;;*****************************************************************
7131  034546  000004                    TST101: SCOPE
7132
7133
7134  034550  012706  001000                    MOV     #STACK,SP       ;RESET STACK
7135  034554  012737  000101  002032            MOV     #TTNO,@#TSTNM   ;THIS SAVES TEST NUMBER
7136  034562  005037  047314                    CLR     @#NOSYNC        ;SET FLAG SO THAT DATA SYNC
7137                                                                    ;AND DATA IS  READ
7138
7139                                    ;*FILL SIMULATED DISK
7140
7141  034566  004737  044216                    JSR     PC,@#SETDSK     ;SET UP SIMULATED DISK
7142
7143                                    ;*FILL REINTO BUFFER WITH 0
7144
7145  034572  004037  042452                    JSR     R0,@#CLAREA     ;FILL REINTO BUFFER
7146  034576  003154                            REINTO                  ;FROM LOCATION
7147  034600  004154                            REINTO+<256.*2>         ;TO LOCATION
7148  034602  000000                            0                       ;DATA
7149
7150                                    ;*FILL WRFROM WITH 10000,401,1,1, AND ALL 177400
7151
7152  034604  012700  002110                    MOV     #WRFROM,R0
7153  034610  012720  010000                    MOV     #FMT22,(R0)+    ;10000 INTO WRFROM
7154  034614  012720  000401                    MOV     #401,(R0)+      ;401=TRACK1,SECTOR1
7155  034620  012720  000001                    MOV     #1,(R0)+        ;1 INTO WRFROM+
7156  034624  012720  000001                    MOV     #1,(R0)+        ;1 INTO WRFROM+6
```

```
7157
7158                                        ;*FILL ALL 0
7159
7160  034630  004037  042452                JSR     R0,@#CLAREA     ;FILL WRFROM
7161  034634  002120                        WRFROM+10               ;FROM
7162  034636  003110                        WRFROM+<256.*2>         ;TO
7163  034640  177400                        177400                  ;DATA
7164
7165                                        ;*NOW GIVE A READ HEADER AND DATA COMMAND
7166                                        ;*CYLINDER=1
7167                                        ;*TRACK = 1
7168                                        ;*SECTOR = 1
7169
7170  034642  004037  044344                JSR     R0,@#HCCRCE
7171  034646  000072                        72                      ;READ HEADER AND DATA
7172  034650  000001                        1                       ;CYLINDER
7173  034652  000001                        1                       ;SECTOR
7174  034654  000001                        1                       ;TRACK
7175  034656  177400                        -256.                   ;WORD COUNT
7176  034660  003154                        REINTO                  ;RHBA BUFFER
7177  034662  000000                        0                       ;READ
7178
7179  034664  000001                        1                       ;HEADER COMPARE
7180  034666  000240                1$:     NOP                     ;RETURN POINT FROM HCCRCE
7191
7192
```

```
                                ;;*****************************************************************
                                ;*TEST 102      MAKE CURRENT CYLINDER = 0
                                ;;*****************************************************************
034670  000004                TST102: SCOPE
034672  012706  001000                MOV     #STACK,SP       ;RESET STACK
034676  012737  000102  002032        MOV     #TTNO,@#TSTNM   ;THIS SAVES TEST NUMBER
034704  004737  042534                JSR     PC,@#CLDISK     ;INIT DRIVE
034710  012777  000001  144742        MOV     #DMD,@RHMR      ;SET DIAGNOSTIC MODE
034716  004037  045106                JSR     R0,@#MAKECYL    ;SUBROUTINE TO GIVE A SEEK
                                                            ;COMMAND FOLOWED BY AN INIT
                                                            ;THIS SHOULD CHANGE RHCC
034722  000000                        0                       ;CHANGE RHCC TO 0


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

                                ;;*****************************************************************
034724  000004                TST103: SCOPE


034726  012706  001000                MOV     #STACK,SP       ;RESET STACK
034732  012737  000103  002032        MOV     #TTNO,@#TSTNM   ;THIS SAVES TEST NUMBER
034740  005037  047314                CLR     @#NOSYNC        ;SET FLAG SO THAT DATA SYNC
                                                            ;AND DATA IS  READ

                                ;*FILL SIMULATED DISK

034744  004737  044216                JSR     PC,@#SETDSK     ;SET UP SIMULATED DISK

                                ;*FILL REINTO BUFFER WITH 0

034750  004037  042452                JSR     R0,@#CLAREA     ;FILL REINTO BUFFER
034754  003154                        REINTO                  ;FROM LOCATION
034756  004154                        REINTO+<256.*2>         ;TO LOCATION
034760  000000                        0                       ;DATA

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

034762  012700  002110                MOV     #WRFROM,R0
034766  012720  010000                MOV     #FMT22,(R0)+    ;10000 INTO WRFROM
034772  012720  000401                MOV     #401,(R0)+      ;401=TRACK1,SECTOR1
034776  012720  000001                MOV     #1,(R0)+        ;1 INTO WRFROM+
035002  012720  000001                MOV     #1,(R0)+        ;1 INTO WRFROM+6
```

```
7239                                        ;*FILL ALL 0
7240
7241  035006  004037  042452                JSR     R0,@#CLAREA     ;FILL WRFROM
7242  035012  002120                        WRFROM+10               ;FROM
7243  035014  003110                        WRFROM+<256.*2>         ;TO
7244  035016  177400                        177400                  ;DATA
7245
7246                                        ;*NOW GIVE A READ HEADER AND DATA COMMAND
7247                                        ;*CYLINDER=0
7248                                        ;*TPACK = 0
7249                                        ;*SECTOR = 1
7250
7251  035020  004037  044344                JSR     R0,@#HCCRCE
7252  035024  000072                        72                      ;READ HEADER AND DATA
7253  035026  000000                        0                       ;CYLINDER
7254  035030  000001                        1                       ;SECTOR
7255  035032  000000                        0                       ;TRACK
7256  035034  177400                        -256.                   ;WORD COUNT
7257  035036  003154                        REINTO                  ;RHBA BUFFER
7259  035040  000000                        0                       ;READ
7259
7260  035042  000001                        1                       ;HEADER COMPARE
7261  035044  000240                1$:     NOP                     ;RETURN POINT FROM HCCRCE
7262
7263
```

```
                                ;;*****************************************************************
                                ;*TEST 104       MAKE CURRENT CYLINDER = 1
                                ;;*****************************************************************
035046  000004                  TST104: SCOPE
035050  012706  001000                  MOV     #STACK,SP       ;RESET STACK
035054  012737  000104  002032          MOV     #TTNO,@#TSTNM   ;THIS SAVES TEST NUMBER
035062  004737  042534                  JSR     PC,@#CLDISK     ;INIT DRIVE
035066  012777  000001  144564          MOV     #DMD,@RHMR      ;SET DIAGNOSTIC MODE
035074  004037  045106                  JSR     R0,@#MAKECYL    ;SUBROUTINE TO GIVE A SEEK
                                                                ;COMMAND FOLOWED BY AN INIT
                                                                ;THIS SHOULD CHANGE RHCC
035100  000001                          1                       ;CHANGE RHCC TO 1


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

                                ;;*****************************************************************
035102  000004                  TST105: SCOPE

035104  012706  001000                  MOV     #STACK,SP       ;RESET STACK
035110  012737  000105  002032          MOV     #TTNO,@#TSTNM   ;THIS SAVES TEST NUMBER

035116  012737  177777  047314          MOV     #-1,@#NOSYNC    ;SET FLAG SO THAT DATA SYNC
                                                                ;AND DATA IS NOT READ

                                ;*FILL SIMULATED DISK

035124  004737  044216                  JSR     PC,@#SETDSK     ;SETUP SIMULATED DISK

                                ;*FILL WRFROM WITH 125252

035130  004037  042452                  JSR     R0,@#CLAREA     ;FILL WRFROM BUFFER
035134  002110                          WRFROM                  ;FROM LOCATION
035136  003110                          WRFROM+<256.*2>         ;TO LOCATION
035140  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

035142  004037  042452                  JSR     R0,@#CLAREA     ;FILL REINTO BUFFER
035146  003154                          REINTO                  ;FROM LOCATION
035150  004154                          REINTO+<256.*2>         ;TO
```

```
 7320  035152  177400                          177400
 7321
 7322                                          ;*NOW GIVE A WRITE DATA COMMAND
 7323                                          ;*CYLINDER = 1,
 7324                                          ;*TRACK = 1
 7325                                          ;*SECTOR        = 1
 7326
 7327  035154  004037  044344                  JSR     R0,@#HCCRCE
 7328  035160  000060                          60                      ;WRITE DATA
 7329  035162  000001                          1                       ;CYLINDER
 7330  035164  000001                          1                       ;SECTOR
 7331  035166  000001                          1                       ;TRACK
 7332  035170  177400                          -256.                   ;WORD COUNT
 7333  035172  002110                          WRFROM                  ;RHPA BUFFER
 7334  035174  000001                          1                       ;WRITE
 7335
 7336  035176  000001                          1                       ;HEADER COMPARE
 7337  035200  000240                  1$:     NOP                     ;RETURN POINT FROM HCCRCE
 7338
```

```
                                ;;*****************************************************************
                                ;*TEST 106       MAKE CURRENT CYLINDER = 0
                                ;;*****************************************************************
035202  000004                  TST106: SCOPE
035204  012706  001000                  MOV     #STACK,SP       ;RESET STACK
035210  012737  000106  002032          MOV     #TTNO,@#TSTNM   ;THIS SAVES TEST NUMBER
035216  004737  042534                  JSR     PC,@#CLDISK     ;INIT DRIVE
035222  012777  000001  144430          MOV     #DMD,@RHMR      ;SET DIAGNOSTIC MODE
035230  004037  045106                  JSR     R0,@#MAKECYL    ;SUBROUTINE TO GIVE A SEEK
                                                                ;COMMAND FOLOWED BY AN INIT
                                                                ;THIS SHOULD CHANGE RHCC
035234  000000                          0                       ;CHANGE RHCC TO 0


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

                                ;;*****************************************************************
035236  000004                  TST107: SCOPE


035240  012706  001000                  MOV     #STACK,SP       ;RESET STACK
035244  012737  000107  002032          MOV     #TTNO,@#TSTNM   ;THIS SAVES TEST NUMBER

035252  012737  177777  047314          MOV     #-1,@#NOSYNC    ;SET FLAG SO THAT DATA SYNC
                                                                ;AND DATA IS NOT READ

                                ;*FILL SIMULATED DISK

035260  004737  044216                  JSR     PC,@#SETDSK     ;SETUP SIMULATED DISK

                                ;*FILL WRFROM WITH 125252

035264  004037  042452                  JSR     R0,@#CLAREA     ;FILL WRFROM BUFFER
035270  002110                          WRFROM                  ;FROM LOCATION
035272  003110                          WRFROM+<256.*2>         ;TO LOCATION
035274  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

035276  004037  042452                  JSR     R0,@#CLAREA     ;FILL REINTO BUFFER
```

```
 7395  035302  003154                        REINTO                  ;FROM LOCATION
 7396  035304  004154                        REINTO+<256.*2>         ;TO
 7397  035306  177400                        177400
 7398
 7399                                        ;*NOW GIVE A WRITE DATA COMMAND
 7400                                        ;*CYLINDER = 0,
 7401                                        ;*TRACK = 0
 7402                                        ;*SECTOR        = 1
 7403
 7404  035310  004037  044344                JSR     R0,@#HCCRCE
 7405  035314  000060                        60                      ;WRITE DATA
 7406  035316  000000                        0                       ;CYLINDER
 7407  035320  000001                        1                       ;SECTOR
 7408  035322  000000                        0                       ;TRACK
 7409  035324  177400                        -256.                   ;WORD COUNT
 7410  035326  002110                        WRFPOM                  ;RHBA BUFFER
 7411  035330  000001                        1                       ;WRITE
 7412
 7413  035332  000001                        1                       ;HEADER COMPARE
 7414  035334  000240                1$:     NOP                     ;RETURN POINT FROM HCCRCE
 7415
```

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

                                ;;*****************************************************************
035336  000004                  TST110: SCOPE

035340  012706  001000                  MOV     #STACK,SP       ;RESET STACK
035344  012737  000110  002032          MOV     #TTNO,@#TSTNM   ;THIS SAVES TEST NUMBER
035352  005037  047314                  CLR     @#NOSYNC        ;SET FLAG SO THAT DATA SYNC
                                                                ;AND DATA IS  READ

                                ;*FILL SIMULATED DISK

035356  004737  044216                  JSR     PC,@#SETDSK     ;SET UP SIMULATED DISK
035362  005137  051100                  COM     @#WCRC          ;CHANGE CRC TO GIVE HCRC

                                ;*FILL REINTO BUFFER WITH 0

035366  004037  042452                  JSR     R0,@#CLAREA     ;FILL REINTO BUFFER
035372  003154                  REINTO                          ;FROM LOCATION
035374  004154                  REINTO+<256.*2>                 ;TO LOCATION
035376  000000                  0                               ;DATA

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

035400  012700  002110                  MOV     #WRFROM,R0
035404  012720  010000                  MOV     #FMT22,(R0)+    ;10000 INTO WRFROM
035410  012720  000401                  MOV     #401,(R0)+      ;401=TRACK1,SECTOR1
035414  012720  000001                  MOV     #1,(R0)+        ;1 INTO WRFROM+
035420  012720  000001                  MOV     #1,(R0)+        ;1 INTO WRFROM+6

                                ;*FILL ALL 0

035424  004037  042452                  JSR     R0,@#CLAREA     ;FILL WRFROM
035430  002120                  WRFROM+10                       ;FROM
035432  003110                  WRFROM+<256.*2>                 ;TO
035434  177400                  177400                          ;DATA

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

035436  004037  044344                  JSR     R0,@#HCCRCE
```

```
 7472  035442  000072                        72                      ;READ HEADER AND DATA
 7473  035444  000000                        0                       ;CYLINDER
 7474  035446  000001                        1                       ;SECTOR
 7475  035450  000001                        1                       ;TRACK
 7476  035452  177400                        -256.                   ;WORD COUNT
 7477  035454  003154                        REINTO                  ;RHBA BUFFER
 7478  035456  000000                        0                       ;READ
 7479
 7480  035460  000000                        0                       ;CRC ERROR
 7481  035462  000240                1$:     NOP                     ;RETURN POINT FROM HCCRCE
 7482
 7483
```

```
                                ;;*****************************************************************
                                ;*TEST 111      RHER1 - BIT 8 - CRC ERROR (WRITING)

                                ;**     THE SIMULATED DISK IS SET UP FOR CYLINDER=0, TRACK=1
                                ;**     SECTOR=1, HEYS=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

                                ;;*****************************************************************
035464  000004                  TST111: SCOPE


035466  012706  001000                  MOV     #STACK,SP       ;RESET STACK
035472  012737  000111  002032          MOV     #TTNO,@#TSTNM   ;THIS SAVES TEST NUMBER

035500  012737  177777  047314          MOV     #-1,@#NOSYNC    ;SET FLAG SO THAT DATA SYNC
                                                                ;AND DATA IS NOT READ

                                ;*FILL SIMULATED DISK

035506  004737  044216                  JSR     PC,@#SETDSK     ;SETUP SIMULATED DISK
035512  005137  051100                  COM     @#WCRC          ;CHANGE CRC TO GIVE HCRC

                                ;*FILL WRFROM WITH 125252

035516  004037  042452                  JSR     R0,@#CLAREA     ;FILL WRFROM BUFFER
035522  002110                          WRFROM                  ;FROM LOCATION
035524  003110                          WRFROM+<256.*2>         ;TO LOCATION
035526  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

035530  004037  042452                  JSR     R0,@#CLAREA     ;FILL REINTO BUFFER
035534  003154                          REINTO                  ;FROM LOCATION
035536  004154                          REINTO+<256.*2>         ;TO
035540  177400                          177400

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

035542  004037  044344                  JSR     R0,@#HCCRCE
035546  000060                          60                      ;WRITE DATA
035550  000000                          0                       ;CYLINDER
035552  000001                          1                       ;SECTOR
035554  000001                          1                       ;TRACK
035556  177400                          -256.                   ;WORD COUNT
```

```
7540  035560  002110                    WRFROM                  ;RHBA BUFFER
7541  035562  000001                    1                       ;WRITE
7542
7543  035564  000000                    0                       ;CRC ERROR
7544  035566  000240              1$:   NOP                     ;RETURN POINT FROM HCCRCE
7545
7546
```

```
7547
7548                                    ;*SET UP FOR THE TWO LAST SECTOR TRANSFERRED TESTS FOLLOWING
7549
7550                                    ;;*****************************************************************
7551  035570  005737  002036            TST     @#RP06  ;MOVE RP06 FLAG TO ITSELF TO TEST
7552  035574  001401                    BEQ     2$              ;IF = 0 TREAT DRIVE AS RP04
7553  035576  000402                    BR      3$              ;TREAT AS RP06 - DO NEXT "MAKECL" & TEST
7554  035600  000137  036254            2$:     JMP     @#DOG           ;DO SECOND FOLLOWING "MAKECL" AND TEST
7555  035604                            3$:
7556                                    ;;*****************************************************************
7557
7558
7559
7560                                    ;;*****************************************************************
7561                                    ;*TEST 112       MAKE CURRENT CYLINDER = 814.
7562                                    ;;*****************************************************************
7563  035604  000004                    TST112: SCOPE
7564  035606  012706  001000                    MOV     #STACK,SP       ;RESET STACK
7565  035612  012737  000112  002032            MOV     #TTNO,@#TSTNM   ;THIS SAVES TEST NUMBER
7566  035620  004737  042534                    JSR     PC,@#CLDISK     ;INIT DRIVE
7567  035624  012777  000001  144026            MOV     #DMD,@RHMR      ;SET DIAGNOSTIC MODE
7568  035632  004037  045106                    JSR     R0,@#MAKECYL    ;SUBROUTINE TO GIVE A SEEK
7569                                                                    ;COMMAND FOLOWED BY AN INIT
7570                                                                    ;THIS SHOULD CHANGE RHCC
7571  035636  001456                            814.                    ;CHANGE RHCC TO 814.
7572
7573
7574
7575
7576
7577
7578                                    ;;*****************************************************************
7579                                    ;*TEST 113       RHDS1 (BIT #10) - LAST SECTOR TRANSFERRED, "LST"
7580
7581                                    ;**     WRITE CYLINDER 814., FORMAT 16 BITS PER WORD
7582                                    ;**     TRACK 18., SECTOR 21., KEYS 0, NUMBER OF WORDS
7583                                    ;**     256., OF 377
7584                                    ;**     "LST" BIT # 10 - RHDS1, SHOULD SET AFTER WRITE
7585                                    ;**     IS COMPLETE.
7586
7587                                    ;;*****************************************************************
7588  035640  000004                    TST113: SCOPE
7589
7590
7591  035642  012706  001000                    MOV     #STACK,SP       ;RESET STACK
7592  035646  012737  000113  002032            MOV     #TTNO,@#TSTNM   ;THIS SAVES TEST NUMBER
7593
7594  035654  004037  042452                    JSR     R0,@#CLAREA     ;CLEAR SIMULATED DISK
7595  035660  051116                            .WORD   DISK            ;FROM
7596  035662  052142                            .WORD   TOLGAP+16       ;TO
7597  035664  000000                            .WORD   0               ;DATA
7598
7599                                            ;*THESE ARE SETUP FOR DISKLESS USE ONLY
7600
7601  035666  012737  011456  047200            MOV     #814.!FMT22,@#CYL;CYLINDER 814.
7602                                                                    ;16 BITS PER WORD
```

C16

```
7603  035674  112737  000022  047203          MOVB    #18.,@#SECOTR+1;TRACK 18.
7604  035702  112737  000025  047202          MOVB    #21.,@#SECOTR   ;SECTOR 21.
7605  035710  005037  047204                  CLR     @#KEY1          ;KEY1 0
7606  035714  005037  047206                  CLR     @#KEY2          ;KEY2 0
7607  035720  012737  000400  047246          MOV     #256.,@#NOWORD  ;NO OF DATA WORDS
7608  035726  012737  000001  047210          MOV     #1,@#X          ;WRITE DATA
7609  035734  004537  043742                  JSR     R5,@#CRC        ;GO TO CALCULATE CRC
7610  035740  047200                          CYL
7611  035742  051100                          WCRC
7612
7613                                          ;*THESE ARE REGULAR SETUPS
7614
7615  035744  004037  042452                  JSR     R0,@#CLAREA     ;FILL WRITE BUFFER WITH 377
7616  035750  002110                          WRFROM                  ;FROM LOCATION
7617  035752  003110                          WRFROM+<256.*2>         ;TO LOCATION
7618  035754  000377                          377                     ;DATA
7619  035756  004737  042534                  JSR     PC,@#CLDISK     ;SETUP GENERAL REGISTERS
7620  035762  012777  177400  143642          MOV     #-256.,@RHWC    ;256. DATA WORDS
7621  035770  012777  002110  143636          MOV     #WRFROM,@RHBA   ;STARTING ADDRESS OF WRITE BUFFER
7622  035776  012746  000025                  MOV     #21.,-(SP)      ;SECTOR 21.
7623  036002  112766  000022  000001          MOVB    #18.,1(SP)      ;TRACK 18.
7624  036010  012677  143630                  MOV     (SP)+,@RHDST    ;SECTOR 21. TRACK 18.
7625  036014  012777  010000  143626          MOV     #FMT22,@RHOF    ;16 BITS PER WORD FORMAT
7626  036022  012777  001456  143622          MOV     #814.,@PHCA     ;CYLINDER 814.
7627  036030  004737  042570                  JSR     PC,@#CHECKT     ;CHECK THAT DVA,RDY,DPR,DRY = 1
7628  036034  104401  062145                  TYPE    ,CPHALT         ;AND THAT NO OTHERS = 1.  CANNOT CON-
7629                                                                  ;TINUE TESTING IF BOTH AREN'T TRUE
7630  036040  000000                          HALT                    ;STOP THE TEST
7631  036042  013711  002062                  MOV     @#WRIDAT,@R1    ;WRITE DATA=60
7632  036046  005037  002006                  CLR     @#ERFLG$        ;CLEAR ERROR FLAG
7633  036052  004737  047040                  JSR     PC,@#COMHD      ;WRITE DATA
7634
7635
7636                                  ;*IF THE PROGRAM COMES BACK HERE WITHOUT ERROR PRINTOUTS
7637                                  ;*FROM THE "COMHD" ROUTINE IT MEANS SECTOR GAP, SYNC BYTE
7638                                  ;*HEADER, HEADER CRC, HEADER GAP AND SYNC BYTE HAVE GONE BY
7639                                  ;*AND SYNCS WERE CORRECTLY DETECTED, DATA IS TO BE CHECKED.
7640
7641  036056  004737  042140                  JSR     PC,@#PUTREG     ;SAVE REGISTERS
7642  036062  005737  002006                  TST     @#ERFLG$        ;HAVE ANY ERRORS OCCURED?
7643  036066  001062                          BNE     5$              ;BRANCH IF YES
7644  036070  012700  000377                  MOV     #377,R0         ;GOOD DATA
7645  036074  012701  051116                  MOV     #DISK,R1        ;DATA WRITTEN INTO "DISK"
7646  036100  012702  000400                  MOV     #256.,R2        ;COUNTER
7647
7648  036104  012737  000401  047320  1$:     MOV     #256.+1,@#ERWORD;FOR ERROR WORD
7649  036112  020021                          CMP     R0,(R1)+        ;COMPARE GOOD DATA WITH DATA ON DISK
7650  036114  001424                          BEQ     3$              ;BRANCH IF GOOD
7651  036116  010037  001124                  MOV     R0,@#$GDDAT     ;GOOD DATA
7652  036122  014137  001126                  MOV     -(R1),@#$BDDAT  ;BAD DATA
7653  036126  160237  047320                  SUB     R2,@#ERWORD     ;ERROR WORD NO
7654  036132  005737  002006                  TST     @#ERFLG$        ;ANY ERRORS ALREADY THERE?
7655  036136  001002                          BNE     2$              ;BRANCH IF YES
7656  036140  104004                          ERROR   4               ;ERROR ON WRITE DATA COMMAND
7657  036142  000401                          BR      64$             ;BRANCH TO AVOID PRINTING NEXT ERROR
7658
```

```
7659  036144  104005                  2$:   ERROR  5                ;WORD NO GIVES WORD IN ERROR
7660  036146  005721                  64$:  TST    (R1)+             ;UNDO -(R1) FOR BAD DATA
7661  036150  017746  142764          	MOV    @SWR,-(SP)        ;GET SWITCH SETTING
7662  036154  042716  177177          	BIC    #177177,(SP)      ;KEEP ONLY SWITCH 7 AND 8
7663  036160  022726  000200          	CMP    #SW07,(SP)+       ;IS 7 SET AND 8 RESET
7664  036164  001402                  	BEQ    4$                ;BRANCH OUT IF YES
7665  036166  005302                  3$:   DEC    R2                ;IF NOT COUNT 256 WORDS
7666  036170  001345                  	BNE    1$                ;BRANCH IF 256. NOT DONE
7667
7668
7669  036172  013746  001736          4$:   MOV    @#DS1,-(SP)       ;GET RHDS1
7670  036176  042716  001000          	BIC    #PROG,(SP)        ;CLEAR PROG
7671  036202  022726  002700          	CMP    #LSTIDPRIDRYIVV,(SP)+;IS "LST" HIGH ?
7672  036206  001412                  	BEQ    5$                ;BRANCH IF GOOD
7673  036210  013737  001662  042204  	MOV    @#RHDS1,@#REGADR ;FAILING REG. ADDRESS
7674  036216  012737  002700  001124  	MOV    #LSTIDPRIDRYIVV,@#$GDDAT ;GOOD DATA
7675  036224  013737  001736  001126  	MOV    @#DS1,@#$BDDAT   ;BAD DATA
7676  036232  104001                  	ERROR  1                ;"LST" DID NOT SET AFTER
7677                                  	                        ;LAST SECTOR ON LAST TRACK
7678                                  	                        ;ON LAST CYLINDER WAS
7679                                  	                        ;WRITTEN
7680                                  	                        ;VV BIT #6 MAY OR MAY NOT BE HIGH
7681  036234  013737  001640  036244  5$:   MOV    @#RHCS1,@#6$     ;SET UP "WAT" SUBROUTINE
7682  036242  104415                  	WAT
7683  036244  000000                  6$:   0                       ;RHCS1 ADDRESS
7684  036246  000200                  	RDY                     ;WAIT FOR READY
7685
7686  036250  000137  036720          	JMP    @#CAT            ;DON'T DO THE RP04 "LST" TEST FOLLOWING
7687
```

```
7688
7689
7690
7691  036254                                DOG:
7692                                        ;;*****************************************************************
7693                                        ;*TEST 114      MAKE CURRENT CYLINDER = 410.
7694                                        ;;*****************************************************************
7695  036254  000004                        TST114: SCOPE
7696  036256  012706  001000                        MOV     #STACK,SP       ;RESET STACK
7697  036262  012737  000114  002032                MOV     #TTNO,@#TSTNM   ;THIS SAVES TEST NUMBER
7698  036270  004737  042534                        JSR     PC,@#CLDISK     ;INIT DRIVE
7699  036274  012777  000001  143356                MOV     #DMD,@RHMR      ;SET DIAGNOSTIC MODE
7700  036302  004037  045106                        JSR     R0,@#MAKECYL    ;SUBROUTINE TO GIVE A SEEK
7701                                                                        ;COMMAND FOLOWED BY AN INIT
7702                                                                        ;THIS SHOULD CHANGE RHCC
7703  036306  000632                        410.                            ;CHANGE RHCC TO 410.
7704
7705
7706
7707
7708                                        ;;*****************************************************************
7709                                        ;*TEST 115      RHDS1 (BIT #10) - LAST SECTOR TRANSFERRED, 'LST'
7710
7711                                        ;**     WRITE CYLINDER 410., FORMAT 16 BITS PER WORD
7712                                        ;**     TRACK 18., SECTOR 21., KEYS 0, NUMBER OF WORDS
7713                                        ;**     256., OF 377
7714                                        ;**     'LST' BIT #10 - RHDS1 SHOULD SET AFTER THE
7715                                        ;**     WRITE IS COMPLETED
7716
7717                                        ;;*****************************************************************
7718  036310  000004                        TST115: SCOPE
7719
7720
7721  036312  012706  001000                        MOV     #STACK,SP       ;RESET STACK
7722  036316  012737  000115  002032                MOV     #TTNO,@#TSTNM   ;THIS SAVES TEST NUMBER
7723
7724  036324  004037  042452                        JSR     R0,@#CLAREA     ;CLEAR SIMULATED DISK
7725  036330  051116                                .WORD   DISK            ;FROM
7726  036332  052142                                .WORD   TOLGAP+16       ;TO
7727  036334  000000                                .WORD   0               ;DATA
7728
7729                                        ;*THESE ARE SETUP FOR DISKLESS USE ONLY
7730
7731  036336  012737  010632  047200                MOV     #410.!PMT22,@#CYL;CYLINDER 410.
7732                                                                        ;16 BITS PER WORD
7733  036344  112737  000022  047203                MOVB    #18.,@#SECOTR+1;TRACK 18.
7734  036352  112737  000025  047202                MOVB    #21.,@#SECOTR   ;SECTOR 21.
7735  036360  005037  047204                        CLR     @#KEY1          ;KEY1 0
7736  036364  005037  047206                        CLR     @#KEY2          ;KEY2 0
7737  036370  012737  000400  047246                MOV     #256.,@#NOWORD  ;NO OF DATA WORDS
7738  036376  012737  000001  047210                MOV     #1,@#X          ;WRITE DATA
7739  036404  004537  043742                        JSR     R5,@#CRC        ;GO TO CALCULATE CRC
7740  036410  047200                                CYL
7741  036412  051100                                WCRC
7742
7743                                        ;*THESE ARE REGULAR SETUPS
```

```
7744
7745  036414  004037  042452                  JSR     R0,@#CLAREA     ;FILL WRITE BUFFER WITH 377
7746  036420  002110                          WRFROM                  ;FROM LOCATION
7747  036422  003110                          WRFROM+<256.*2>         ;TO LOCATION
7748  036424  000377                          377                     ;DATA
7749  036426  004737  042534                  JSR     PC,@#CLDISK     ;SETUP GENERAL REGISTERS
7750  036432  012777  177400  143172          MOV     #-256.,@RHWC    ;256. DATA WORDS
7751  036440  012777  002110  143166          MOV     #WRFROM,@RHBA   ;STARTING ADDRESS OF WRITE BUFFER
7752  036446  012746  000025                  MOV     #21.,-(SP)      ;SECTOR 21.
7753  036452  112766  000022  000001          MOVB    #18.,1(SP)      ;TRACK 18.
7754  036460  012677  143160                  MOV     (SP)+,@RHDST    ;SECTOR 21. TRACK 18.
7755  036464  012777  010000  143156          MOV     #FMT22,@RHOF    ;16 BITS PER WORD FORMAT
7756  036472  012777  000632  143152          MOV     #410.,@RHCA     ;CYLINDER 410.
7757  036500  004737  042570                  JSR     PC,@#CHECKT     ;CHECK THAT DVA,RDY,DPR,DRY = 1
7758  036504  104401  062145                  TYPE    ,CPHALT         ;AND THAT NO OTHERS = 1.  CANNOT CON-
7759                                                                  ;TINUE TESTING IF BOTH AREN'T TRUE
7760  036510  000000                          HALT                    ;STOP THE TEST
7761  036512  013711  002062                  MOV     @#WRIDAT,@R1    ;WRITE DATA=60
7762  036516  005037  002006                  CLR     @#ERFLG$        ;CLEAR ERROR FLAG
7763  036522  004737  047040                  JSR     PC,@#COMHD      ;WRITE DATA
7764
7765
7766                                  ;*IF THE PROGRAM COMES BACK HERE WITHOUT ERROR PRINTOUTS
7767                                  ;*FROM THE "COMHD" ROUTINE IT MEANS SECTOR GAP, SYNC BYTE
7768                                  ;*HEADER, HEADER CRC, HEADER GAP AND SYNC BYTE HAVE GONE BY
7769                                  ;*AND SYNCS WERE CORRECTLY DETECTED, DATA IS TO BE CHECKED.
7770
7771  036526  004737  042140                  JSR     PC,@#PUTREG     ;SAVE REGISTERS
7772  036532  005737  002006                  TST     @#ERFLG$        ;HAVE ANY ERRORS OCCURED?
7773  036536  001062                          BNE     5$              ;BRANCH IF YES
7774  036540  012700  000377                  MOV     #377,R0         ;GOOD DATA
7775  036544  012701  051116                  MOV     #DISK,R1        ;DATA WRITTEN INTO "DISK"
7776  036550  012702  000400                  MOV     #256.,R2        ;COUNTER
7777
7778  036554  012737  000401  047320  1$:     MOV     #256.+1,@#ERWORD;FOR ERROR WORD
7779  036562  020021                          CMP     R0,(R1)+        ;COMPARE GOOD DATA WITH DATA ON DISK
7780  036564  001424                          BEQ     3$              ;BRANCH IF GOOD
7781  036566  010037  001124                  MOV     R0,@#$GDDAT     ;GOOD DATA
7782  036572  014137  001126                  MOV     -(R1),@#$BDDAT  ;BAD DATA
7783  036576  160237  047320                  SUB     R2,@#ERWORD     ;ERROR WORD NO
7784  036602  005737  002006                  TST     @#ERFLG$        ;ANY ERRORS ALREADY THERE?
7785  036606  001002                          BNE     2$              ;BRANCH IF YES
7786  036610  104004                          ERROR   4               ;ERROR ON WRITE DATA COMMAND
7787  036612  000401                          BR      64$             ;BRANCH TO AVOID PRINTING NEXT ERROR
7788
7789  036614  104005          2$:     ERROR   5               ;WORD NO GIVES WORD IN ERROR
7790  036616  005721          64$:    TST     (R1)+           ;UNDO -(R1) FOR BAD DATA
7791  036620  017746  142314          MOV     @SWR,-(SP)      ;GET SWITCH SETTING
7792  036624  042716  177177          BIC     #177177,(SP)    ;KEEP ONLY SWITCH 7 AND 8
7793  036630  022726  000200          CMP     #SW07,(SP)+     ;IS 7 SET AND 8 RESET
7794  036634  001402                  BEQ     4$              ;BRANCH OUT IF YES
7795  036636  005302          3$:     DEC     R2              ;IF NOT COUNT 256 WORDS
7796  036640  001345                  BNE     1$              ;BRANCH IF 256. NOT DONE
7797
7798
7799  036642  013746  001736          4$:     MOV     @#DS1,-(SP)     ;GET RHDS1
```

```
7800  036646  042716  001000                  BIC     #PROG,(SP)          ;CLEAR PROG BIT
7801  036652  022726  002700                  CMP     #LST!DPR!DRY!VV,(SP)+  ;IS "LST" HIGH ?
7802  036656  001412                          BEQ     5$                  ;WAIT FOR "RDY" IF GOOD
7803  036660  013737  001662  042204          MOV     @#RHDS1,@#REGADR ;FAILING REG. ADDRESS
7804  036666  012737  002700  001124          MOV     #LST!DPR!DRY!VV,@#$GDDAT ;GOOD DATA
7805  036674  013737  001736  001126          MOV     @#DS1,@#$BDDAT  ;BAD DATA
7806  036702  104001                          ERROR   1               ;"LST" DID NOT SET AFTER
7807                                                                  ;LAST SECTOR ON LAST TRACK ON LAST
7808                                                                  ;CYLINDER WAS WRITTEN - "VV" BIT #6
7809                                                                  ;MAY OR MAY NOT BE HIGH
7810
7811  036704  013737  001640  036714  5$:     MOV     @#RHCS1,@#6$    ;SET UP "WAT" SUBROUTINE
7812  036712  104415                          WAT
7813  036714  000000                  6$:     0                       ;RHCS1 ADDRESS
7814  036716  000200                          RDY                     ;WAIT FOR "RDY" BIT
```

```
                                CAT:
                                ;;*******************************************************************
                                ;*TEST 116      ERROR REGISTER 1 - BIT #9 AOE
                                ;**     A WRITE DATA COMMAND IS GIVEN TO CYLINDER 410./914.
                                ;**     SECTOR 21 TRACK 18, KEYS 0, DATA 377
                                ;**     WORD COUNT REGISTER FOR 326 (256+66+4) WORDS
                                ;**
                                ;**     AFTER 256 WORDS HAVE BEEN WRITTEN
                                ;**     'AOE' SHOULD COME UP
                                ;**     RHWC WILL SHOW 4 BECAUSE THE SILO IS 66 WORDS AND
                                ;**     256 WORDS HAVE BEEN WRITTEN - TOTAL 322
                                ;**     THIS IS 4 SHORT OF 326
                                ;;*******************************************************************
036720  000004                  TST116: SCOPE
036722  012706  001000                  MOV     #STACK,SP       ;RESET STACK
036726  012737  000116  002032          MOV     #TTWO,@#TSTNM   ;THIS SAVES TEST NUMBER
036734  004737  042534                  JSR     PC,@#CLDISK     ;INIT AND SET UP GENERAL REG. CORRES.
036740  004037  042452                  JSR     R0,@#CLAREA     ;CLEAR SIMULATED DISK
036744  051116                          .WORD   DISK            ;FROM
036746  052142                          .WORD   TOLGAP+16       ;TO
036750  000000                          .WORD   0               ;DATA

                                ;*THESE ARE TO SETUP FOR DISKLESS USE ONLY
                                ;*AND WILL HANDLE RP04 OR RP06 DRIVES

036752  005737  002036          TST     @#RP06  ;MOVE RP06 FLAG TO ITSELF TO TEST
036756  001404                  BEQ     10$             ;TREAT DRIVE AS RP04 IF = 0

036760  012737  011456  047200  MOV     #814.!FMT22,@#CYL;CYLINDER 814., 16 BITS PER WORD
036766  000403                  BR      11$             ;TREAT DRIVE AS RP06

036770  012737  010632  047200  10$:    MOV     #410.!FMT22,@#CYL;CYLINDER 410., 16 BITS PER WORD
                                                        ;TREAT DRIVE AS RP04

036776  112737  000022  047203  11$:    MOVB    #18.,@#SECOTR+1 ;TRACK 18.
037004  112737  000025  047202          MOVB    #21.,@#SECOTR   ;SECTOR 21.
037012  005037  047204                  CLR     @#KEY1          ;KEY1 0
037016  005037  047206                  CLR     @#KEY2          ;KEY2 0
037022  012737  000400  047246          MOV     #256.,@#NOWORD  ;NO OF DATA WORDS
037030  012737  000001  047210          MOV     #1,@#X          ;WRITE DATA
037036  004537  043742                  JSR     R5,@#CRC        ;GO TO CALCULATE CRC
037042  047200                          CYL
037044  051100                          WCRC

                                ;*THESE ARE REGULAR SETUPS

037046  004037  042452                  JSR     R0,@#CLAREA     ;FILL WRITE BUFFER WITH 377
037052  002110                          WRFROM                  ;FROM
037054  003110                          WRFROM+<256.*2>         ;TO
037056  000377                          377                     ;DATA
037060  004737  042534                  JSR     PC,@#CLDISK     ;SETUP GENERAL REGISTERS
```

```
7871  037064  012777  177272  142540          MOV     #-326.,@RHWC    ;326. DATA WORDS
7872  037072  012777  002110  142534          MOV     #WRFROM,@PHBA   ;STARTING ADDRESS OF WRITE BUFFER
7873  037100  012746  000025                  MOV     #21.,-(SP)      ;SECTOR 21.
7874  037104  112766  000022  000001          MOVB    #18.,1(SP)      ;TRACK 18.
7875  037112  012677  142526                  MOV     (SP)+,@RHDST    ;SECTOR 21. TRACK 18.
7876  037116  012777  010000  142524          MOV     #FMT22,@RHOF    ;16 BITS PER WORD FORMAT
7877
7878                                          ;*CHECK TO SEE WHAT TYPE OF DRIVE IS BEING TESTED
7879                                          ;*AND LOAD CYLINDER ADDRESS REGISTER WITH THE PROPER NUMBER
7880
7881  037124  005737  002036                  TST     @#RP06  ;MOVE FLAG TO ITSELF TO TEST
7882  037130  001404                          BEQ     12$             ;TREAT AS RP04 IF = 0
7883  037132  012777  001456  142512          MOV     #814.,@PHCA     ;CYLINDER 814.
7884  037140  000403                          BR      13$             ;TREAT AS RP06
7885  037142  012777  000632  142502  12$:    MOV     #410.,@RHCA     ;CYLINDER 410.
7886  037150                          13$:
7887
7888  037150  004737  042570                  JSR     PC,@#CHECKT     ;CHECK THAT DVA,RDY,DPR,DRY = 1
7889  037154  104401  062145                  TYPE    ,CPHALT         ;AND THAT NO OTHERS = 1.  CANNOT CON-
7890                                                                  ;TINUE TESTING IF BOTH AREN'T TRUE
7891  037160  000000                          HALT                    ;STOP THE TEST
7892  037162  013711  002062                  MOV     @#WRIDAT,@R1    ;WRITE DATA=60
7893  037166  005037  002006                  CLR     @#ERFLG$        ;CLEAR ERROR FLAG
7894
7895                                          ;*THE REGISTERS WILL BE SAVED IN REINTO BUFFER
7896
7897  037172  004037  043226                  JSR     R0,@#SAVER      ;SAVE
7898  037176  001632                          RHWC                    ;FROM
7899  037200  003154                          REINTO                  ;TO
7900  037202  000023                          19.                     ;NUMBER SAVED
7901
7902                                          ;*GIVE WRITE DATA COMMAND
7903
7904  037204  004737  047040                  JSR     PC,@#COMND      ;WRITE DATA COMMAND
7905
7906                                          ;*CHANGE SAVED REGISTERS TO EXPECTED VALUES
7907
7908  037210  005737  002040                  TST     @#RH70          ;CHECK FOR RH70 CONTROLLER
7909  037214  001407                          BEQ     8$              ;SKIP RH70 CODE AND DO RH11 IF NOT
7910
7911  037216  012737  177702  003154          MOV     #-76,@#REINTO   ;SAVED RHWC SHOULD BE = 76 (OCTAL)
7912  037224  012737  003130  003156          MOV     #WRFROM+<2*256.>+<2*8.>,@#REINTO+2
7913                                                                  ;SAVED RHBA SHOULD BE WRFROM+256+8
7914  037232  000406                          BR      9$              ;SKIP NEXT RH11 CODE
7915
7916  037234  012737  177774  003154  8$:     MOV     #-4,@#REINTO    ;SAVED RHWC SHOULD BE = 4
7917  037242  012737  003314  003156          MOV     #WRFROM+<2*256.>+<2*66.>,@#REINTO+2
7918                                                                  ;SAVED RHBA SHOULD BE WRFROM+256+66
7919
7920  037250  052737  000200  003160  9$:     BIS     #OR,@#REINTO+4  ;SAVED RHCS2
7921  037256  042737  000100  003160          BIC     #IR,@#REINTO+4  ;SAVED RHCS2
7922  037264  052737  140000  003162          BIS     #SCITRE,@#REINTO+6;SAVED RHCS1 SHOULD HAVE 'SC' & 'TRE'
7923  037272  012737  001000  003164          MOV     #AOE,@#REINTO+10 ;SAVED RHER1 SHOULD HAVE 'AOE'
7924  037300  017737  142340  003166          MOV     @RHDST,@#REINTO+12;SAVED RHDST SHOULD HAVE=
7925                                                                  ;RHDST IS UNDEFINED
7926
```

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

037306  005737  002036                TST     @#RP06  ;MOVE RP06 FLAG TO ITSELF TO TEST
037312  001404                        BEQ     14$             ;TREAT AS RP04 IF = 0
037314  012737  001457  003174        MOV     #815.,@#REINTO+20;SAVED DESIRED CYLINDER ADDRESS
037322  000403                        BR      15$             ;TREAT AS RP06
037324  012737  000633  003174  14$:  MOV     #411.,@#REINTO+20;SAVED DESIRED CYLINDER ADDRESS

037332  013737  002016  003200  15$:  MOV     @#ATTENT,@#REINTO+24 ;SAVED RHAS SHOULD HAVE APPRO. BIT
037340  052737  000001  003202        BIS     #DMD,@#REINTO+26;SAVED RHMR
037346  052737  142000  003204        BIS     #ATA!ERRILST,@#REINTO+30 ;SAVED RHDS1

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

037354  004037  043226                JSR     R0,@#SAVER      ;SAVE
037360  001632                        RHWC                    ;FROM
037362  002110                        WRFROM                  ;TO
037364  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

037366  113737  003201  002135        MOVB    @#REINTO+25,@#WRFROM+25;SAVE UPPER RHAS

                                        ;*COMPARE REGISTERS BEFORE WRITE DATA COMMAND
                                        ;*WITH AFTER COMMAND

037374  004037  043430                JSR     R0,@#COMPAR     ;COMPARE
037400  003154                        REINTO                  ;GOOD BUFFER
037402  002110                        WRFROM                  ;TEST BUFFER
037404  000021                        17.                     ;NUMBER OF REGISTERS
037406  037414                        1$                      ;RETURN FOR ERROR
037410  037414                        1$                      ;SAME
037412  037434                        2$                      ;RETURN FOR GOOD COMPARISON

037414  013705  047320          1$:   MOV     @#ERWORD,R5     ;GETTING READY TO INDEX
037420  060505                        ADD     R5,R5           ;DOUBLE ERROR WORD
037422  016537  001630  042204        MOV     RHWC-2(R5),@#REGADR ;FAILING REG. ADDRESS
037430  104001                        ERROR   1               ;FORCED AOE ERROR CAUSED IMPROPER
                                                                ;REGISTER CHANGE
037432  000207                        RTS     PC              ;RETURN FOR FURTHER COMPARISONS
                                                                ;NO ERRORS
037434  005037  002006          2$:   CLR     @#ERFLG$        ;CLEAR ERROR FLAG

                                        ;*DATA IS TO BE CHECKED HERE

037440  004737  042140                JSR     PC,@#PUTREG     ;SAVE REGISTERS
037444  012700  000377                MOV     #377,R0         ;GOOD DATA
037450  012701  051116                MOV     #DISK,R1        ;DATA WRITTEN INTO "DISK"
037454  012702  000400                MOV     #256.,R2        ;COUNTER
037460  012737  000400  047320  3$:   MOV     #256.,@#ERWORD  ;FOR ERROR WORD
037466  020021                        CMP     R0,(R1)+        ;COMPARE GOOD DATA WITH DATA ON DISK
```

K15

```
 7983  037470  001424                        BEQ     6$              ;BRANCH IF GOOD
 7984  037472  010037  001124                MOV     R0,@#$GDDAT     ;GOOD DATA
 7985  037476  014137  001126                MOV     -(R1),@#$BDDAT  ;BAD DATA
 7986  037502  160237  047320                SUB     R2,@#ERWORD     ;ERROR WORD NO
 7987  037506  005737  002006                TST     @#ERFLG$        ;ANY ERRORS ALREADY THERE?
 7988  037512  001002                        BNE     4$              ;BRANCH IF YES
 7989  037514  104004                        ERROR   4               ;ERROR ON WRITE DATA COMMAND WITH FORCED "AOE"
 7990  037516  000401                        BR      5$              ;BRANCH TO AVOID PRINTING NEXT ERROR
 7991  037520  104005              4$:       ERROR   5               ;WORD NO. GIVES WORD IN ERROR
 7992  037522  005721              5$:       TST     (R1)+           ;UNDO -(R1) FOR BAD DATA
 7993  037524  017746  141410                MOV     @SWR,-(SP)      ;GET SWITCH SETTING
 7994  037530  042716  177177                BIC     #177177,(SP)    ;KEEP ONLY SWITCH 7 AND 8
 7995  037534  022726  000200                CMP     #SW07,(SP)+     ;IS 7 SET AND 8 RESET
 7996  037540  001402                        BEQ     7$              ;BRANCH OUT IF YES ------------------->
 7997
 7998  037542  005302              6$:       DEC     R2              ;IF NOT COUNT 256 WORDS
 7999  037544  001345                        BNE     3$              ;BRANCH IF 256. NOT DONE
 8000
 8001  037546              7$:
```

```
8002
8003                                    ;;*****************************************************************
8004                                    ;*TEST 117      MAKE CURRENT CYLINDER = 0
8005                                    ;;*****************************************************************
8006  037546  000004                    TST117: SCOPE
8007  037550  012706  001000                    MOV     #STACK,SP       ;RESET STACK
8008  037554  012737  000117  002032            MOV     #TTNO,@#TSTNM   ;THIS SAVES TEST NUMBER
8009  037562  004737  042534                    JSR     PC,@#CLDISK     ;INIT DRIVE
8010  037566  012777  000001  142064            MOV     #DMD,@RHMR      ;SET DIAGNOSTIC MODE
8011  037574  004037  045106                    JSR     R0,@#MAKECYL    ;SUBROUTINE TO GIVE A SEEK
8012                                                                    ;COMMAND FOLOWED BY AN INIT
8013                                                                    ;THIS SHOULD CHANGE RHCC
8014  037600  000000                            0                       ;CHANGE RHCC TO 0
8015
8016
8017
8018                                    ;;*****************************************************************
8019                                    ;*TEST 120      ERROR REGISTER 1 - BIT #10 'IAE'
8020
8021                                    ;**     A READ HEADER AND DATA IS GIVEN TO TRACK 20, SECTOR 0
8022                                    ;**
8023                                    ;**     AN INDEX PULSE IS GIVEN TO GET RHLA TO 0
8024                                    ;**
8025                                    ;**     IAE BIT SHOULD SET
8026
8027                                    ;;*****************************************************************
8028  037602  000004                    TST120: SCOPE
8029  037604  012706  001000                    MOV     #STACK,SP       ;RESET STACK
8030  037610  012737  000120  002032            MOV     #TTNO,@#TSTNM   ;THIS SAVES TEST NUMBER
8031  037616  004737  042534                    JSR     PC,@#CLDISK     ;CLEAR REGISTERS AND SET UNIT NO.
8032
8033                                            ;*GIVE INDEX PULSE
8034  037622  012777  000001  142030            MOV     #DMD,@RHMR      ;SET DIAGNOSTIC MODE
8035  037630  052777  000004  142022            BIS     #MINX,@RHMR     ;SET INDEX
8036  037636  042777  000004  142014            BIC     #MINX,@RHMR     ;CLEAR INDEX
8037
8038
8039                                            ;*THESE ARE REGULAR SETUPS
8040
8041  037644  012777  177400  141760            MOV     #-256.,@RHWC    ;256 DATA  WORDS 4 HEADER WORDS
8042  037652  012700  003154                    MOV     #REINTO,R0      ;THESE TWO INSTRUCTIONS GETS
8043  037656  010077  141752                    MOV     R0,@RHBA        ;ADDR, OF WRFROM INTO R0 AND
8044                                                                    ;BUS ADDRESS REGISTER
8045  037662  012720  010000                    MOV     #FMT22,(R0)+    ;FORMAT=16 BIT WORDS
8046                                                                    ;CYLINDER=0
8047  037666  012720  012000                    MOV     #12000,(R0)+    ;TRACK=20 SECTOR=0 KEYS=0
8048  037672  005020                            CLR     (R0)+           ;KEY1=0
8049  037674  005020                            CLR     (R0)+           ;KEY2=0
8050  037676  012705  000400                    MOV     #256.,R5        ;COUNTER
8051  037702  012720  177777            1$:     MOV     #-1,(R0)+       ;MOVE ALL ONES FOR DATA
8052  037706  005305                            DEC     R5
8053  037710  001374                            BNE     1$              ;BRANCH IF DATA NOT COMPLETE
8054  037712  012777  012000  141724            MOV     #12000,@RHDST   ;TRACK=20 SECTOR=0
8055
8056  037720  004737  042570                    JSR     PC,@#CHECKT     ;CHECK THAT DVA,RDY,DPR,DRY = 1
8057  037724  104401  062145                    TYPE    ,CPHALT         ;AND THAT NO OTHERS = 1.  CANNOT CON-
```

m15

```
8058                                                    ;TINUE TESTING IF BOTH AREN'T TRUE
8059  037730  000000                        HALT        ;STOP THE TEST
8060
8061  037732  013711  002070                MOV   @#REFOR,@R1   ;GET READY FOR WRITE HEADER AND
8062                                                    ;DATA WITH 62 IN RHCS1
8063  037736  005037  002006                CLR   @#ERFLG$      ;CLEAR ERROR FLAG
8064  037742  012777  010000  141700        MOV   #FMT22,@RHOF  ;FORMAT BIT=1 (16 BIT WORDS)
8065  037750  005077  141676                CLR   @RHCA         ;CYLINDER =0
8066
8067                                        ;*THE REGISTERS WILL BE SAVED IN REINTO BUFFER
8068  037754  004037  043226                JSR   R0,@#SAVER    ;SAVE
8069  037760  001632                        RHWC                ;FROM
8070  037762  003154                        REINTO              ;TO
8071  037764  000023                        19.                 ;NUMBER SAVED
8072
8073                                        ;*GO TO WRITE HEADER AND DATA
8074
8075  037766  013700  001660                MOV   @#RHMR,R0     ;NOW R0 WAS MAINTENANCE REG. ADDR.
8076  037772  012710  000001                MOV   #DMD,@R0      ;SET DIAGNOSTIC MODE
8077  037776  052777  000001  141634        BIS   #GO,@RHCS1    ;GO
8078
8079                                        ;*CHANGE SAVED REGISTERS TO EXPECTED VALUE
8080  040004  052737  140000  003162        BIS   #SCITRE,@#REINTO+6 ;SAVED RHCS1
8081  040012  012737  002000  003164        MOV   #IAE,@#REINTO+10 ;SAVED RHER1
8082  040020  012737  012001  003166        MOV   #12001,@#REINTO+12;SAVED RHDST
8083  040026  013737  002016  003200        MOV   @#ATTENT,@#REINTO+24 ;SAVED RHAS
8084  040034  052737  000001  003202        BIS   #DMD,@#REINTO+26 ;SAVED RHMR
8085  040042  052737  140000  003204        BIS   #ATAIERR,@#REINTO+30 ;SAVED RHDS1
8086
8087                                        ;*SAVE REGISTERS AGAIN SO THAT COMPARES CAN BE DONE
8088  040050  004037  043226                JSR   R0,@#SAVEP    ;SAVE
8089  040054  001632                        RHWC                ;FROM
8090  040056  002110                        WRFROM              ;TO
8091  040060  000023                        19.                 ;NUMBER OF REGISTERS SAVED
8092
8093                                        ;*AS UPPER BYTE OF RHAS CAN BE CHANGING IN A DUAL PORT
8094                                        ;*OPERATION THE UPPER BYTE OF RHAS WILL BE SAVED AS IS
8095                                        ;*SO THAT THE COMPARES ARE ONLY VALID FOR THE LOWER BYTE
8096  040062  113737  003201  002135        MOVB  @#REINTO+25,@#WRFROM+25;SAVE UPPER RHAS
8097
8098
8099                                        ;*COMPARE REGISTERS BEFORE READ IN PRESET COMMAND
8100                                        ;*WITH AFTER COMMAND
8101  040070  004037  043430                JSR   R0,@#COMPAR   ;COMPARE
8102  040074  003154                        REINTO              ;GOOD BUFFER
8103  040076  002110                        WRFROM              ;TEST BUFFER
8104  040100  000021                        17.                 ;NUMBER OF REGISTERS
8105  040102  040110                        2$                  ;RETURN FOR ERROR
8106  040104  040110                        2$                  ;SAME
8107  040106  040130                        3$                  ;RETURN FOR GOOD COMPARISON
8108
8109  040110  013705  047320        2$:     MOV   @#ERWORD,R5   ;GETTING READY TO INDEX
8110  040114  060505                        ADD   R5,R5         ;DOUBLE ERROR WORD
8111  040116  016537  001630  042204        MOV   RHMC-2(R5),@#REGADR ;FAILING REG. ADDRESS
8112  040124  104001                        ERROR 1             ;FORCED IAE CAUSED IMPROPER
8113                                                            ;REGISTER CHANGE
```

```
 8114  040126  000207                          RTS     PC              ;RETURN FOR FURTHER COMPARISONS
 8115
 8116                                                                  ;NO ERRORS
 8117
 8118  040130  004737  042534          3$:     JSR     PC,@#CLDISK     ;CLEAR GO BIT
 8119
```

```
8120
8121                                    ;;*****************************************************************
8122                                    ;*TEST 121      ERROR REGISTER 1- BIT #10 'IAE'
8123
8124                                    ;**     A WRITE HEADER AND DATA IS GIVEN TO SECTOR 22
8125                                    ;**     TRACK 0 CYLINDER 0
8126                                    ;**
8127                                    ;**     WORD COUNT IS SET TO 256.
8128                                    ;**
8129                                    ;**     AN INDEX PULSE IS GIVEN TO GET RHLA TO 0
8130                                    ;**
8131                                    ;**     IAE BIT SHOULD SET
8132
8133                                    ;;*****************************************************************
8134  040134  000004                   TST121: SCOPE
8135  040136  012706  001000                   MOV     #STACK,SP       ;RESET STACK
8136  040142  012737  000121  002032           MOV     #TTWO,@#TSTNM   ;THIS SAVES TEST NUMBER
8137  040150  004737  042534                   JSR     PC,@#CLDISK     ;CLEAR REGISTERS AND SET UNIT NO.
8138
8139                                           ;*GIVE INDEX PULSE
8140  040154  012777  000001  141476           MOV     #DMD,@RHMP      ;SET DIAGNOSTIC MODE
8141  040162  052777  000004  141470           BIS     #MINX,@RHMR     ;SET INDEX
8142  040170  042777  000004  141462           BIC     #MINX,@RHMR     ;CLEAR INDEX
8143
8144                                           ;*THESE ARE REGULAR SETUPS
8145
8146  040176  012777  177400  141426           MOV     #-256.,@RHWC    ;256 DATA WORDS 4 HEADER WORDS
8147  040204  012700  002110                   MOV     #WRFROM,R0      ;THESE TWO INSTRUCTIONS GETS
8148  040210  010077  141420                   MOV     R0,@RHBA        ;ADDR. OF WRFROM INTO R0 AND
8149                                                                   ;BUS ADDRESS REGISTER
8150  040214  012720  010000                   MOV     #FMT22,(R0)+    ;FORMAT=16 BIT WORDS
8151                                                                   ;CYLINDER=0
8152  040220  012720  000026                   MOV     #22.,(R0)+      ;TRACK=0, SECTOR=22, KEYS=0
8153  040224  005020                           CLR     (R0)+           ;KEY1=0
8154  040226  005020                           CLR     (R0)+           ;KEY2=0
8155  040230  012705  000400                   MOV     #256.,R5        ;COUNTER
8156  040234  012720  177777           1$:     MOV     #-1,(R0)+       ;MOVE ALL ONES FOR DATA
8157  040240  005305                           DEC     R5
8158  040242  001374                           BNE     1$              ;BRANCH IF DATA NOT COMPLETE
8159  040244  012777  000026  141372           MOV     #22.,@RHDST     ;TRACK=0 SECTOR=22
8160
8161  040252  004737  042570                   JSR     PC,@#CHECKT     ;CHECK THAT DVA,RDY,DPR,DRY = 1
8162  040256  104401  062145                   TYPE    ,CPHALT         ;AND THAT NO OTHERS = 1.  CANNOT CON-
8163                                                                   ;TINUE TESTING IF BOTH AREN'T TRUE
8164  040262  000000                           HALT                    ;STOP THE TEST
8165
8166  040264  013711  002064                   MOV     @#WRIFOR,@R1    ;GET READY FOR WRITE HEADER AND
8167                                                                   ;DATA WITH 62 IN RHCS1
8168  040270  005037  002006                   CLR     @#ERFLG$        ;CLEAR ERROR FLAG
8169  040274  012777  010000  141346           MOV     #FMT22,@RHOF    ;FORMA BIT=1 (16 BIT WORDS)
8170  040302  005077  141344                   CLR     @RHCA           ;CYLINDER =0
8171
8172                                    ;*AS EXCEPTION IS ASSERTED BEFORE RUN IS
8173                                    ;*LATCHED RHWC,RHBA,RHCS1,RHCS2 CANNOT BE CHECKED
8174                                    ;*BECAUSE RHWC WILL VARY DEPENDING UPON GATE DELAYS
8175                                    ;*ON DIFFERENT UNITS
```

C16

```
8176
8177                                        ;*THE REGISTERS WILL BE SAVED IN REINTO BUFFER
8178  040306  004037  043226                JSR     R0,@#SAVER      ;SAVE
8179  040312  001642                        RHER1                   ;FROM
8180  040314  003154                        REINTO                  ;TO
8181  040316  000015                        13.                     ;NUMBER SAVED
8182
8183                                        ;*GO TO WRITE HEADER AND DATA
8184
8185  040320  013700  001660                MOV     @#RHMR,R0       ;NOW R0 HAS MAINTENANCE REG. ADDR.
8186  040324  012710  000001                MOV     #DMD,@R0        ;SET DIAGNOSTIC MODE
8187  040330  052777  000001  141302        BIS     #GO,@RHCS1      ;GO
8188
8189                                        ;*CHANGE SAVED REGISTERS TO EXPECTED VALUE
8190  040336  012737  002000  003154        MOV     #IAE,@#REINTO   ;SAVED RHER1
8191  040344  012737  000027  003156        MOV     #23.,@#REINTO+2;SAVED RHDST
8192  040352  013737  002016  003170        MOV     @#ATTENT,@#REINTO+14 ;SAVED RHAS
8193  040360  052737  000001  003172        BIS     #DMD,@#REINTO+16 ;SAVED RHMR
8194  040366  052737  140000  003174        BIS     #ATAIERR,@#REINTO+20 ;SAVED RHDS1
8195
8196                                        ;*SAVE REGISTERS AGAIN SO THAT COMPARES CAN BE DONE
8197  040374  004037  043226                JSR     R0,@#SAVER      ;SAVE
8198  040400  001642                        RHER1                   ;FROM
8199  040402  002110                        WRFROM                  ;TO
8200  040404  000015                        13.                     ;NUMBER OF REGISTERS SAVED
8201
8202                                        ;*AS UPPER BYTE OF RHAS CAN BE CHANGING IN A DUAL PORT
8203                                        ;*OPERATION THE UPPER BYTE OF RHAS WILL BE SAVED AS IS
8204                                        ;*SO THAT THE COMPARES ARE ONLY VALID FOR THE LOWER BYTE
8205  040406  113737  003171  002125        MOVB    @#REINTO+15,@#WRFROM+15;SAVE UPPER RHAS
8206
8207
8208                                        ;*COMPARE REGISTERS BEFORE READ IN PRESET COMMAND
8209                                        ;*WITH AFTER COMMAND
8210  040414  004037  043430                JSR     R0,@#COMPAR     ;COMPARE
8211  040420  003154                        REINTO                  ;GOOD BUFFER
8212  040422  002110                        WRFROM                  ;TEST BUFFER
8213  040424  000015                        13.                     ;NUMBER OF REGISTERS
8214  040426  040434                        2$                      ;RETURN FOR ERROR
8215  040430  040434                        2$                      ;SAME
8216  040432  040454                        3$                      ;RETURN FOR GOOD COMPARISON
8217
8218  040434  013705  047320        2$:     MOV     @#ERWORD,R5     ;GETTING READY TO INDEX
8219  040440  060505                        ADD     R5,R5           ;DOUBLE ERROR WORD
8220  040442  016537  001640  042204        MOV     RHER1-2(R5),@#REGADR ;FAILING REG. ADDRESS
8221  040450  104001                        ERROR   1               ;FORCED IAE CAUSED IMPROPER
8222                                                                ;REGISTER CHANGE
8223  040452  000207                        RTS     PC              ;RETURN FOR FURTHER COMPARISONS
8224
8225                                                                ;NO ERRORS
8226
8227  040454  004737  042534        3$:     JSR     PC,@#CLDISK     ;CLEAR GO BIT
8228
8229
```

```
8230
8231                                    ;;*****************************************************************
8232                                    ;*TEST 122      ERROR REGISTER 1- BIT #10 'IAE'
8233
8234                                    ;**     A WRITE DATA IS GIVEN TO SECTOR 0
8235                                    ;**     TRACK 0 CYLINDER 411
8236                                    ;**
8237                                    ;**     WORD COUNT IS SET TO 256.
8238                                    ;**     AN INDEX PULSE IS GIVEN TO GET PHLA TO 0
8239                                    ;**
8240                                    ;**
8241                                    ;**     IAE BIT SHOULD SET
8242
8243                                    ;;*****************************************************************
8244  040460  000004                   TST122: SCOPE
8245  040462  012706  001000                   MOV     #STACK,SP       ;RESET STACK
8246  040466  012737  000122  002032           MOV     #TTNO,@#TSTNM   ;THIS SAVES TEST NUMBER
8247  040474  004737  042534                   JSR     PC,@#CLDISK     ;CLEAR REGISTERS AND SET UNIT NO.
8248
8249                                           ;*GIVE INDEX PULSE
8250  040500  012777  000001  141152           MOV     #DMD,@RHMR      ;SET DIAGNOSTIC MODE
8251  040506  052777  000004  141144           BIS     #MINX,@RHMR     ;SET INDEX
8252  040514  042777  000004  141136           BIC     #MINX,@RHMR     ;CLEAR INDEX
8253
8254                                           ;*THESE ARE REGULAR SETUPS
8255
8256  040522  012777  177400  141102           MOV     #-256.,@RHWC    ;256 DATA WORDS 4 HEADER WORDS
8257  040530  012700  002110                   MOV     #WRFROM,R0      ;THESE TWO INSTRUCTIONS GETS
8258  040534  010077  141074                   MOV     R0,@RHBA        ;ADDR. OF WRFROM INTO R0 AND
8259                                                                   ;BUS ADDRESS REGISTER
8260  040540  012705  000400                   MOV     #256.,R5        ;COUNTER
8261  040544  012720  177777           1$:     MOV     #-1,(R0)+       ;MOVE ALL ONES FOR DATA
8262  040550  005305                           DEC     R5
8263  040552  001374                           BNE     1$              ;BRANCH IF DATA NOT COMPLETE
8264  040554  012777  000000  141062           MOV     #0.,@PHDST      ;TRACK=0 SECTOR=0
8265
8266  040562  004737  042570                   JSR     PC,@#CHECKT     ;CHECK THAT DVA,RDY,DPR,DRY = 1
8267  040566  104401  062145                   TYPE    ,CPHALT         ;AND THAT NO OTHERS = 1.  CANNOT CON-
8268                                                                   ;TINUE TESTING IF BOTH AREN'T TRUE
8269  040572  000000                           HALT                    ;STOP THE TEST
8270
8271  040574  013711  002062                   MOV     @#WRIDAT,@R1    ;GET READY FOR WRITE
8272                                                                   ;DATA WITH 60 IN RHCS1
8273  040600  005037  002006                   CLR     @#ERFLG$        ;CLEAR ERROR FLAG
8274  040604  012777  010000  141036           MOV     #FMT22,@RHOF    ;FORMA BIT=1 (16 BIT WORDS)
8275
8276  040612  005737  002036                   TST     @#RP06  ;MOVE FLAG TO ITSELF TO TEST
8277  040616  001404                           BEQ     4$              ;TREAT DRIVE AS RP04 IF FLAG = 0
8278
8279  040620  012777  001457  141024           MOV     #815.,@PHCA     ;CYLINDER = 815 (ONE TOO MANY)
8280  040626  000403                           BR      5$              ;TREAT DRIVE AS RP06
8281
8282  040630  012777  000633  141014   4$:     MOV     #411.,@PHCA     ;CYLINDER = 411 (ONE TOO MANY)
8283                                                                   ;TREAT DRIVE AS RP04
8284
8285                                           ;*AS EXCEPTION IS ASSERTED BEFORE RUN IS
```

```
8286                                        ;*LATCHED RHWC,RHBA,RHCS1,RHCS2 CANNOT BE CHECKED
8287                                        ;*BECAUSE RHWC WILL VARY DEPENDING UPON GATE DELAYS
8288                                        ;*ON DIFFERENT UNITS
8289
8290                                        ;*THE REGISTERS WILL BE SAVED IN REINTO BUFFER
8291  040636  004037  043226          5$:   JSR     R0,@#SAVER      ;SAVE
8292  040642  001642                        RHER1                   ;FROM
8293  040644  003154                        REINTO                  ;TO
8294  040646  000015                        13.                     ;NUMBER SAVED
8295
8296                                        ;*GO TO WRITE HEADER AND DATA
8297
8298  040650  013700  001660                MOV     @#RHMR,R0       ;NOW R0 HAS MAINTENANCE REG. ADDR.
8299  040654  012710  000001                MOV     #DMD,@R0        ;SET DIAGNOSTIC MODE
8300  040660  052777  000001  140752        BIS     #GO,@RHCS1      ;GO
8301
8302                                        ;*CHANGE SAVED REGISTERS TO EXPECTED VALUE
8303  040666  012737  002000  003154        MOV     #IAE,@#REINTO   ;SAVED RHER1
8304  040674  012737  000001  003156        MOV     #1.,@#REINTO+2;SAVED RHDST
8305  040702  013737  002016  003170        MOV     @PATTENT,@#REINTO+14 ;SAVED RHAS
8306  040710  052737  000001  003172        BIS     #DMD,@#REINTO+16 ;SAVED RHMR
8307  040716  052737  140000  003174        BIS     #ATA!ERR,@#REINTO+20 ;SAVED RHDS1
8308
8309                                        ;*SAVE REGISTERS AGAIN SO THAT COMPARES CAN BE DONE
8310  040724  004037  043226                JSR     R0,@#SAVER      ;SAVE
8311  040730  001642                        RHER1                   ;FROM
8312  040732  002110                        WRFROM                  ;TO
8313  040734  000015                        13.                     ;NUMBER OF REGISTERS SAVED
8314
8315                                        ;*AS UPPER BYTE OF RHAS CAN BE CHANGING IN A DUAL PORT
8316                                        ;*OPERATION THE UPPER BYTE OF RHAS WILL BE SAVED AS IS
8317                                        ;*SO THAT THE COMPARES ARE ONLY VALID FOR THE LOWER BYTE
8318  040736  113737  003171  002125        MOVB    @#REINTO+15,@#WRFROM+15;SAVE UPPER RHAS
8319
8320
8321                                        ;*COMPARE REGISTERS BEFORE READ IN PRESET COMMAND
8322                                        ;*WITH AFTER COMMAND
8323  040744  004037  043430                JSR     R0,@#COMPAR     ;COMPARE
8324  040750  003154                        REINTO                  ;GOOD BUFFER
8325  040752  002110                        WRFROM                  ;TEST BUFFER
8326  040754  000015                        13.                     ;NUMBER OF REGISTERS
8327  040756  040764                        2$                      ;RETURN FOR ERROR
8328  040760  040764                        2$                      ;SAME
8329  040762  041004                        3$                      ;RETURN FOR GOOD COMPARISON
8330
8331  040764  013705  047320          2$:   MOV     @#ERWORD,R5     ;GETTING READY TO INDEX
8332  040770  060505                        ADD     R5,R5           ;DOUBLE ERROR WORD
8333  040772  016537  001640  042204        MOV     RHER1-2(R5),@#REGADR ;FAILING REG. ADDRESS
8334  041000  104001                        ERROR   1               ;FORCED IAE CAUSED IMPROPER
8335                                                                ;REGISTER CHANGE
8336  041002  000207                        RTS     PC              ;RETURN FOR FURTHER COMPARISONS
8337
8338                                                                ;NO ERRORS
8339
8340  041004  004737  042534          3$:   JSR     PC,@#CLDISK     ;CLEAR GO BIT
```

```
                                ;;*****************************************************************
                                ;*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
                                ;
                                ;;*****************************************************************
041010  000004                  TST123: SCOPE
041012  012737  000001  001212          MOV     #1,$TIMES       ;;DO 1 ITERATION
041020  012737  000000  177776          MOV     #0,PS           ;REINSTATE PS TO 0
041026  104401  041034                  TYPE    ,65$            ;;TYPE ASCIZ STRING
041032  000425                          BR      64$             ;;GET OVER THE ASCIZ
                                ;;65$:  .ASCIZ  <15><12>/TOTAL ERRORS ON THIS PASS ON  UNIT NO. /
041106                          64$:
041106  013746  001774                  MOV     @#UNIT,-(SP)    ;GET READY TO TYPE UNIT NUMBER
041112  104405                          TYPDS
041114  104401  041122                  TYPE    ,67$            ;;TYPE ASCIZ STRING
041120  000402                          BR      66$             ;;GET OVER THE ASCIZ
                                ;;67$:  .ASCIZ  /= /
041126                          66$:
041126  013746  001112                  MOV     @#$ERTTL,-(SP)  ;GET READY TO TYPE NUMBER OF ERRORS
041132  104405                          TYPDS
041134  005037  001112                  CLR     @#$ERTTL        ;CLEAR TOTAL NUMBER OF ERRORS
041140  005037  001102                  CLR     @#$TSTNM        ;CLEAR TEST NUMBER
041144  005737  002002                  TST     @#SELECT        ;STARTING FROM 210 ?

041150  001413                          BEQ     3$              ;TEST NEXT DRIVE IF NOT
                                                                ;CONTINUE ON THIS ONE IF SO
041152  005237  001100                  INC     @#$PASS         ;INCREASE PASS COUNT
041156  104401  041341                  TYPE    ,$ENDMG         ;TYPE END PASS #
041162  013746  001100                  MOV     @#$PASS,-(SP)
041166  104405                          TYPDS
041170  104401  041336                  TYPE    ,$ENULL
041174  000137  007460                  JMP     @#TST5          ;DO NEXT TESTS ---------------------->

041200  005337  001776          3$:     DEC     @#NOUNITS       ;NO. OF UNITS PRESENT DECREMENTED
041204  001413                          BEQ     $EOP            ;BRANCH IF ALL DRIVES COMPLETE
041206  013700  001774                  MOV     @#UNIT,R0       ;UNIT UNDER TEST
041212  012701  001754                  MOV     #UNITS,R1       ;TABLE
041216  022100                  1$:     CMP     (R1)+,R0        ;IS THIS UNIT JUST TESTED
041220  001401                          BEQ     2$              ;BRANCH IF YES
041222  000775                          BR      1$              ;BRANCH IF NO
041224  011137  001774          2$:     MOV     (R1),@#UNIT     ;THIS IS NEXT UNIT
041230  000137  007460                  JMP     @#TST5          ;GO FOR NEXT TESTS ------------------->


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


                                .SBTTL  END OF PASS ROUTINE

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

```
 8397                                ;*INCREMENT THE PASS NUMBER ($PASS)
 8398                                ;*TYPE "END PASS #XXXXX" (WHERE XXXXX IS A DECIMAL NUMBER)
 8399                                ;*IF THERES A MONITOR GO TO IT
 8400                                ;*IF THERE ISN'T JUMP TO TST1
 8401
 8402  041234                        $EOP:
 8403  041234  000004                       SCOPE
 8404  041236  005037  001102               CLR     $TSTNM          ;;ZERO THE TEST NUMBER
 8405  041242  005037  001212               CLR     $TIMES          ;;ZERO THE NUMBER OF ITERATIONS
 8406  041246  005237  001100               INC     $PASS           ;;INCREMENT THE PASS NUMBER
 8407  041252  042737  100000  001100       BIC     #100000,$PASS   ;;DON'T ALLOW A NEG. NUMBER
 8408  041260  005327                       DEC     (PC)+           ;;LOOP?
 8409  041262  000001                $EOPCT: .WORD   1
 8410  041264  003022                       BGT     $DOAGN          ;;YES
 8411  041266  012737                       MOV     (PC)+,@(PC)+    ;;RESTORE COUNTER
 8412  041270  000001                $ENDCT: .WORD   1
 8413  041272  041262                       $EOPCT
 8414  041274  104401  041341               TYPE    ,$ENDMG         ;;TYPE "END PASS #"
 8415  041300  013746  001100               MOV     $PASS,-(SP)     ;;SAVE $PASS FOR TYPEOUT
 8416  041304  104405                       TYPDS                   ;;GO TYPE--DECIMAL ASCII WITH SIGN
 8417  041306  104401  041336               TYPE    ,$ENULL         ;;TYPE A NULL CHARACTER
 8418  041312  013700  000042        $GET42: MOV     @#42,R0         ;;GET MONITOR ADDRESS
 8419  041316  001405                       BEQ     $DOAGN          ;;BRANCH IF NO MONITOR
 8420  041320  000005                       RESET                   ;;CLEAR THE WORLD
 8421  041322  004710                $ENDAD: JSR     PC,(R0)         ;;GO TO MONITOR
 8422  041324  000240                       NOP                     ;;SAVE ROOM
 8423  041326  000240                       NOP                     ;;FOR
 8424  041330  000240                       NOP                     ;;ACT11
 8425  041332                        $DOAGN:
 8426  041332  000137                       JMP     @(PC)+          ;;RETURN
 8427  041334  005500                $RTNAD: .WORD   TST1
 8428  041336     377     377     000  $ENULL: .BYTE   -1,-1,0         ;;NULL CHARACTER STRING
 8429  041341     015  042412  042116  $ENDMG: .ASCIZ  <15><12>/END PASS #/
 8430  041346  050040  051501  020123
 8431  041354  000043
 8432
 8433
 8434                                ;;*****************************************************************
 8435                                ;**HERE IS A DETAILED EXPLAINATION OF HOW THE LOOP ON ERROR WORKS.
 8436                                ;**ON HITTING AN ERROR IF THE LOOP ON ERROR SWITCH IS SET, THE
 8437                                ;**PROGRAM GOES BACK - USUALLY BACK TO THE BEGINNING OF THE TEST.
 8438
 8439                                ;**WHEN THIS OPERATOR SELECTABLE SCOPE LOOP IS USED THEN THE POINT
 8440                                ;**THE PROGRAM GOES BACK TO CAN BE CHANGED.
 8441                                ;**THE RESTRICTIONS TO THE POINT WHERE THE PROGRAM CAN GO ARE: -
 8442                                ;**1.  IT MUST BE WITHIN THE TEST UNDER CONSIDERATION
 8443                                ;**2.  LOOP ON ERROR SWITCH MUST BE SET
 8444                                ;**3.  THE ERROR MUST OCCUR WITHIN THE TEST UNDER CONSIDERATION
 8445                                ;**IF THE ERROR DOES NOT OCCUR WITHIN THE TEST UNDER CONSIDERATION
 8446                                ;**THE PROGRAM WILL REVERT TO NORMAL OPERATION.  HOWEVER, IF LOOP ON
 8447                                ;**TEST SWITCH IS SET AND THIS OPERATOR SELECTABLE SCOPE LOOP IS USED
 8448                                ;**THEN THE PROGRAM WILL LOOP BACK TO THE SELECTED POINT WHEN IT
 8449                                ;**COMES TO THE END OF THE TEST UNDER CONSIDERATION.
 8450                                ;**
 8451                                ;**AFTER LOOPING FOR SOME TIME IF THE LOOP SWITCH IS PUT DOWN THEN
 8452                                ;**NORMAL OPERATION WILL CONTINUE.
```

```
 8453                                ;;*****************************************************************
 8454
 8455
 8456  041356  000000                TESTAD: 0                       ;FIRST ADDRESS OF TEST
 8457
 8458  041360                        OPERSEL:
 8459  041360  005037  177776                CLR     PS              ;MAKE PROCESSOR STATUS ZERO
 8460  041364  104401  041372                TYPE    ,65$            ;;TYPE ASCIZ STRING
 8461  041370  000421                        BR      64$             ;;GET OVER THE ASCIZ
 8462                                ;;65$:  .ASCIZ  <15><12>/THE PROGRAM WAS IN TEST NUMBER /
 8463  041434                        64$:
 8464  041434  013746  002032                MOV     @#TSTNM,-(SP)   ;GET READY TO TYPE TEST
 8465  041440  104402                        TYPOC                   ;NUMBER
 8466  041442  104401  041450                TYPE    ,67$            ;;TYPE ASCIZ STRING
 8467  041446  000414                        BR      66$             ;;GET OVER THE ASCIZ
 8468                                ;;67$:  .ASCIZ  <15><12>/THE LOOP BACK PC WAS /
 8469  041500                        66$:
 8470  041500  013746  001110                MOV     @#$LPERR,-(SP)  ;GET READY TO TYPE LOOP BACK PC
 8471  041504  104402                        TYPOC
 8472  041506  104401  001223                TYPE    ,$CRLF
 8473  041512  104401  041520                TYPE    ,69$            ;;TYPE ASCIZ STRING
 8474  041516  000430                        BR      68$             ;;GET OVER THE ASCIZ
 8475                                ;;69$:  .ASCIZ  <15><12>/SET SWITCH FOR LOOP ON ERROR OR LOOP ON TEST/
 8476  041600                        68$:
 8477  041600  104401  041606                TYPE    ,71$            ;;TYPE ASCIZ STRING
 8478  041604  000430                        BR      70$             ;;GET OVER THE ASCIZ
 8479                                ;;71$:  .ASCIZ  <15><12>/TYPE THE FIRST PC OF THE TEST TO BE LOOPED ON/
 8480  041666                        70$:
 8481  041666  104401  041674                TYPE    ,73$            ;;TYPE ASCIZ STRING
 8482  041672  000422                        BR      72$             ;;GET OVER THE ASCIZ
 8483                                ;;73$:  .ASCIZ  <15><12>/ FOLLOWED BY A CARRIAGE RETURN /<15><12>
 8484  041740                        72$:
 8485  041740  104412                        RDOCT
 8486  041742  062716  000002                ADD     #2,(SP)         ;GET LPADR
 8487  041746  012637  001106                MOV     (SP)+,@#$LPADR
 8488  041752  104401  041760                TYPE    ,75$            ;;TYPE ASCIZ STRING
 8489  041756  000417                        BR      74$             ;;GET OVER THE ASCIZ
 8490                                ;;75$:  .ASCIZ  <15><12>/TYPE THE PC WHERE YOU WANT/
 8491  042016                        74$:
 8492  042016  104401  042024                TYPE    ,77$            ;;TYPE ASCIZ STRING
 8493  042022  000440                        BR      76$             ;;GET OVER THE ASCIZ
 8494                                ;;77$:  .ASCIZ  <15><12>/ THE PROGRAM TO LOOP BACK TO FOLLOWED BY A CARRIAGE RETURN /<15
 8495  042124                        76$:
 8496  042124  104412                        RDOCT
 8497  042126  012637  001110                MOV     (SP)+,@#$LPERR  ;GET LPERR
 8498  042132  013746  001106                MOV     @#$LPADR,-(SP)
 8499  042136  000002                        RTI
 8500
 8501
 8502
 8503                                        .SBTTL  SAVE REGISTERS ROUTINE
 8504
 8505
 8506                                        ;*THIS SAVES THE CONTENTS OF ALL HARDWARE REGISTERS
 8507                                        ;*IN MEMORY LOCATIONS TAGED FROM "WC" TO "EC2"
 8508                                        ;*
```

```
8509                                        ;*
8510                                        ;*THIS IS DONE SO THAT COMPARES ARE DONE WITH SAVED LOCATIONS
8511                                        ;*AND NOT THE REGISTERS THEMSELVES.   THIS WILL MAKE
8512                                        ;*ERROR PRINTOUTS FOR GOOD AND BAD DATA ALWAYS DIFFRENT
8513
8514  042140                        PUTREG:
8515  042140  010046                        MOV     R0,-(SP)        ;;PUSH R0 ON STACK
8516  042142  010146                        MOV     R1,-(SP)        ;;PUSH R1 ON STACK
8517  042144  010246                        MOV     R2,-(SP)        ;;PUSH R2 ON STACK
8518  042146  012700  001632                MOV     #RHWC,R0        ;STARTING ADDRESS OF REG
8519  042152  012701  001706                MOV     #WC,R1          ;STARTING ADDRESS OF WERE SAVED
8520  042156  012702  000023                MOV     #RHCC-RHWC+2/2,R2 ;NUMBER OF REG. INTO R2
8521  042162  013021                10$:    MOV     @(R0)+,(R1)+    ;SAVE HARDWARE REG.
8522  042164  005302                        DEC     R2
8523  042166  001375                        BNE     10$
8524  042170  012602                        MOV     (SP)+,R2        ;;POP STACK INTO R2
8525  042172  012601                        MOV     (SP)+,R1        ;;POP STACK INTO R1
8526  042174  012600                        MOV     (SP)+,R0        ;;POP STACK INTO R0
8527  042176  000207                        RTS     PC
8528
8529
8530
8531
8532                                        .SBTTL  FLOAT 1 AND 0
8533
8534
8535
8536                                ;*FLOAT A ONE AND A ZERO THRU A DESIGNATED REGISTER
8537                                ;*ABSOLUTE ADDRESS OF REG. UNDER TEST IS IN R4
8538
8539  042200  000000                MASK:   0                       ;BITS UNDER TEST
8540  042202  000000                LERR:   0                       ;ERROR HLT ADDRESS
8541  042204  000000                REGADR: 0
8542
8543  042206  012537  042200        BITST:  MOV     (R5)+,  MASK    ;FETCH DATA MASK
8544  042212  012504                        MOV     (R5)+,  R4      ;GET ADDRESS OF REG. UNDER TEST
8545  042214  010437  042204                MOV     R4,     REGADR
8546  042220  010537  042202                MOV     R5,     LERR    ;GET ERROR RETURN ADDR.
8547  042224  062705  000004                ADD     #4,     R5      ;MODIFY RETURN ADDR. TO JUMP OVER RTS
8548  042230  012703  000001                MOV     #1,     R3      ;INITIALIZE DATA PATTERN
8549  042234  004737  042256        BLT1:   JSR     PC,     BLT2    ;OUTPUT FLOATING ZERO
8550  042240  004737  042256                JSR     PC,     BLT2    ;OUTPUT FLOATING ONE
8551  042244  000241                        CLC
8552  042246  006103                        ROL     R3              ;SHIFT PATTERN
8553  042250  005703                        TST     R3
8554  042252  001370                        BNE     BLT1            ;BRANCH IF NOT COMPLETE
8555  042254  000205                        RTS     R5              ;RETURN TO TEST
8556  042256  005103                BLT2:   COM     R3              ;COMPLEMENT PATTERN
8557  042260  012737  042266  042514        MOV     #BLT3,  @#LAD   ;SET SCOPE LOOP
8558  042266  032777  001000  136644  BLT3:   BIT     #SW09,@SWR      ;LOOP ON ERROR
8559  042274  001411                        BEQ     4$              ;BRANCH IF NO
8560  042276  105737  001103                TSTB    @#$ERFLG        ;ANY ERRORS
8561  042302  001406                        BEQ     4$      ;BRANCH IF NO
8562  042304  000005                        RESET                   ;START WITH AN INIT
8563  042306  013777  001774  137322        MOV     @#UNIT,@RHCS2   ;SET UNIT NUMBER UNDER TEST
8564  042314  004737  054672                JSR     PC,@#$TKINT     ;INITILIZE TK
```

```
8565
8566  042320  010337  001124        4$:   MOV   R3,@#$GDDAT     ;STORE GOOD DATA
8567  042324  005137  042200              COM   @#MASK            ;AND MASK WITH PATTERN
8568  042330  043737  042200  001124      BIC   @#MASK, @#$GDDAT        ;CLEAR THE REST
8569  042336  005137  042200              COM   @#MASK            ;RESTORE MASK
8570  042342  013714  001124              MOV   @#$GDDAT,(R4)    ;OUTPUT TO REGISTER
8571  042346  011437  001126              MOV   (R4),@#$BDDAT    ;INPUT FROM REGISTER
8572  042352  005137  042200              COM   @#MASK
8573  042356  043737  042200  001126      BIC   @#MASK,@#$BDDAT ;AND MASK OUT RECEIVED DATA
8574  042364  005137  042200              COM   @#MASK            ;RESTORE MASK
8575  042370  023737  001124  001126      CMP   @#$GDDAT,@#$BDDAT       ;IS DATA CORRECT
8576  042376  001424                      BEQ   1$                ;BRANCH IF GOOD
8577  042400  011437  001126              MOV   (R4),@#$BDDAT
8578  042404  023704  001640              CMP   @#RHCS1,R4        ;REGISTER UNDER TEST RHCS1?
8579  042410  001004                      BNE   2$                ;BRANCH IF NOT
8580  042412  052737  004200  001124      BIS   #RDY!DVA,@#$GDDAT;SET RDY AND DVA
8581  042420  000410                      BR    3$
8582  042422  023704  001636        2$:   CMP   @#RHCS2,R4        ;REGISTER UNDER TEST RHCS2?
8583  042426  001005                      BNE   3$                ;BRANCH IF NOT
8584  042430  011446                      MOV   @R4,-(SP)         ;GET RHCS2
8585  042432  042716  177477              BIC   #^C<IR!OR>,(SP) ;KEEP IR AND OR BIT
8586  042436  052637  001124              BIS   (SP)+,@#$GDDAT  ;SET IR OR BITS IF NEEDED
8587  042442  004777  177534        3$:   JSR   PC,     @LERR    ;GO TO REPORT ERROR
8588  042446  000240                      NOP                     ;REPLACE BY 104420 FOR LOCAL SCOPE LOOP
8589  042450  000207                1$:   RTS   PC
8590
8591
8592                                        .SBTTL  CLEAR MEMORY ROUTINE
8593
8594                                ;*      THIS CLEARS ANY BLOCK OF MEMORY
8595                                ;*      FILLING IT WITH ANY DATA
8596                                ;*
8597                                ;*      CALL
8598                                ;*      JSR     R0,CLAREA
8599                                ;*      X                       ;STARTING ADDRESS OF BLOCK
8600                                ;*      Y
8601                                ;*      Z                       ;DATA TO BE FILLED
8602                                ;*R1 WILL HAVE STARTING ADDRESS OF BLOCK TO BE FILLED
8603                                ;*R2 AFTER SUBTRACTION WILL HAVE TWICE NUMBER OF LOCATIONS
8604                                ;*R3 WILL HAVE DATA TO BE FILLED
8605                                ;*TO AVOID DIVIDE ROUTINE TWO DECREMENT R2 WILL BE USED
8606
8607
8608  042452                        CLAREA:
8609  042452  010146                      MOV   R1,-(SP)          ;;PUSH R1 ON STACK
8610  042454  010246                      MOV   R2,-(SP)          ;;PUSH R2 ON STACK
8611  042456  010346                      MOV   R3,-(SP)          ;;PUSH R3 ON STACK
8612  042460  012001                      MOV   (R0)+,R1          ;FROM
8613  042462  012002                      MOV   (R0)+,R2          ;TO
8614  042464  012003                      MOV   (R0)+,R3          ;DATA
8615  042466  160102                      SUB   R1,R2             ;NO. OF LOCATIONS MINUS TWO
8616  042470  062702  000002              ADD   #2,R2             ;GET TWICE NO OF LOCATIONS
8617  042474  010321                1$:   MOV   R3,(R1)+          ;MOVE IN DATA
8618  042476  005302                      DEC   R2
8619  042500  005302                      DEC   R2
8620  042502  001374                      BNE   1$                ;BRANCH IF NOT COMPLETE
```

K16

```
 8621  042504  012603                    MOV     (SP)+,R3        ;;POP STACK INTO R3
 8622  042506  012602                    MOV     (SP)+,R2        ;;POP STACK INTO R2
 8623  042510  012601                    MOV     (SP)+,R1        ;;POP STACK INTO R1
 8624  042512  000200                    RTS     R0              ;RETURN
```

```
8625                                            .SBTTL  LOCAL TRAPS
8626  042514  000000                    LAD:    0
8627
8628  042516  032777  001000  136414    T.SCOP: BIT     #SW09,  @SWR
8629  042524  001402                            BEQ     1$
8630  042526  013716  042514                    MOV     @#LAD,  (SP)
8631  042532  000002                    1$:     RTI
8632
8633                                    ;*EXAMPLE OF THE USE OF THE ABOVE
8634                                    ;*THIS WILL LOOP BETWEEN X: AND SCOP1 PROVIDED THERE IS NO "NEWTST"
8635                                    ;*MOV   #X,     @#LAD
8636                                    ;*X:    ---     ---
8637                                    ;*      ---     ---
8638                                    ;*      ---     ---
8639                                    ;*      SCOP1
8640
8641
8642                                            .SBTTL  CLEAR DISK ROUTINE
8643
8644  042534  013701  001640            CLDISK: MOV     @#RHCS1,        R1      ;R1 WILL BE CONTROL AND STATUS1
8645  042540  013702  001636                    MOV     @#RHCS2,        R2      ;R2 WILL BE CONTROL AND STATUS2
8646  042544  013703  001662                    MOV     @#RHDS1,        R3      ;R3 WILL BE DISK STATUS REGISTER1
8647  042550  013704  001642                    MOV     @#RHER1,        R4      ;R4 WILL BE ERROR REGISTER #1
8648
8649  042554  012712  000040                    MOV     #CLR,@R2                ;CLEAR ALL REG.
8650  042560  013712  001774                    MOV     @#UNIT,@R2      ;REINSTATE UNIT NO.
8651  042564  005011                            CLR     @R1                     ;CLEAR FUNCTION BITS
8652  042566  000207                            RTS     PC
```

```
                                        .SBTTL  CHECK DISP 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

042570  011637  002014          CHECKT: MOV     (SP),@#PCJSR    ;SAVE PC OF JSR+4
042574  162737  000004  002014          SUB     #4,@#PCJSR      ;GET PC OF JSR
042602  004737  042140                  JSR     PC,@#PUTREG     ;SAVE REGISTERS
042606  022737  004200  001714          CMP     #DVA!RDY,@#CS1  ;RHCS1 SHOULD HAVE DEVICE AVAILABLE
                                                                ;AND BE READY
042614  001423                          BEQ     3$              ;BRANCH IF GOOD

042616  032737  004000  001714          BIT     #DVA,@#CS1      ;BAD SO TEST DEVICE AVAILABLE
042624  001004                          BNE     1$              ;BRANCH IF DVA THERE
042626  010137  001122                  MOV     R1,@#$BDADR     ;ADDRESS OF BAD REGISTER (RHCS1)
042632  104026                          ERROR   26              ;RHCS1 DID NOT HAVE DEVICE
                                                                ;AVAILABLE AT START OF TEST
042634  000413                          BR      3$              ;BRANCH TO NEXT COMPARE
042636  032737  000200  001714  1$:     BIT     #RDY,@#CS1      ;TEST READY
042644  001003                          BNE     2$              ;IF RDY THERE BRANCH
042646  010137  001122                  MOV     R1,@#$BDADR     ;ADDRESS OF BAD REGISTER (RHCS1)
042652  104026                          ERROR   26              ;RHCS1 DID NOT HAVE READY
                                                                ;RIGHT AT START OF TEST
042654  000403                  2$:     BR      3$              ;BRANCH TO NEXT COMPARE
042656  010137  001122                  MOV     R1,@#$BDADR     ;ADDRESS OF BAD REGISTER (RHCS1)
042662  104026                          ERROR   26              ;RHCS1 HAD SOME BITS OTHER
                                                                ;THAN DVA AND RDY SET
                                                                ;ALL OTHER BITS SHOULD BE 0

042664  013746  001736          3$:     MOV     @#DS1,-(SP)     ;GET RHDS1
042670  042716  001100                  BIC     #VV!PROG,(SP)   ;CLEAR VV AND PROGRAMABLE BIT
042674  022726  000600                  CMP     #DPR!DRY,(SP)+;RHDS1 SHOULD HAVE THESE SET
042700  001424                          BEQ     8$              ;BRANCH IF THEY ARE

042702  032737  000400  001736  4$:     BIT     #DPR,@#DS1      ;TEST DRIVE PRESENT
042710  001004                          BNE     5$              ;CONTINUE IF THERE
042712  010337  001122                  MOV     R3,@#$BDADR     ;ADDRESS OF BAD REGISTER (RHDS1)
042716  104026                          ERROR   26              ;RHDS1 DOES NOT HAVE DPR
042720  000413                          BR      7$              ;BRANCH OUT
042722  032737  000200  001736  5$:     BIT     #DRY,@#DS1      ;TEST DRIVE READY
042730  001004                          BNE     6$              ;IF DPR WAS THERE, BRANCH IF GOOD
042732  010337  001122                  MOV     R3,@#$BDADR     ;ADDRESS OF BAD REGISTER (RHDS1)
042736  104026                          ERROR   26              ;RHDS1 DOES NOT HAVE DRY
042740  000403                          BR      7$              ;BRANCH OUT
042742  010337  001122          6$:     MOV     R3,@#$BDADR     ;ADDRESS OF BAD REGISTER (RHDS1)
042746  104026                          ERROR   26              ;RHDS1 HAS SOME BITS OTHER
                                                                ;THAN MOL, DRY, DPR, SET
                                                                ;ALL OTHER BITS SHOULD BE 0
042750  000207                  7$:     RTS     PC              ;RETURN TO TEST AND HALT

042752  062716  000006          8$:     ADD     #6,(SP)         ;ADJUST STACK TO GET OVER HALT IN TEST
042756  000207                          RTS     PC              ;RETURN TO TEST AND CONTINUE TESTING
```

A1

```
 8709
```

```
8710
8711                                    ;*THIS CHECKS THAT DEVICE AVAILABLE (DVA) AND READY (RDY) IN RHCS1 = 1
8712                                    ;*AND CHECKS THAT DEVICE PRESENT (DPR), DEVICE READY (DRY) IN RHDS1 = 1
8713
8714  042760  011637  002014          CHECKE: MOV     (SP),@#PCJSR    ;SAVE PC OF JSR+4
8715  042764  162737  000004  002014          SUB     #4,@#PCJSR      ;GET PC OF JSR
8716  042772  004737  042140                  JSR     PC,@#PUTREG     ;READ & SAVE REGISTERS
8717  042776  032737  000200  001714          BIT     #RDY,@#CS1      ;RHCS1 SHOULD HAVE DEVICE AVAILABLE
8718                                                                  ;AND BE READY
8719  043004  001004                          BNE     1$              ;BRANCH IF GOOD
8720  043006  010137  001122                  MOV     R1,@#$BDADR     ;FAILING REGISTER
8721  043012  104026                          ERROR   26              ;RHCS1 IS IN ERROR
8722                                                                  ;DOES NOT HAVE DVA, RDY
8723  043014  000427                          BR      4$              ;BRANCH OUT
8724  043016  032737  004000  001714  1$:     BIT     #DVA,@#CS1      ;RHCS1 SHOULD HAVE DEVICE AVAILABLE
8725                                                                  ;AND BE READY
8726  043024  001004                          BNE     2$              ;BRANCH IF GOOD
8727  043026  010137  001122                  MOV     R1,@#$BDADR     ;FAILING REGISTER
8728  043032  104026                          ERROR   26              ;RHCS1 IS IN ERROR
8729                                                                  ;DOES NOT HAVE DVA, RDY
8730  043034  000417                          BR      4$              ;BRANCH OUT
8731  043036  032737  000200  001736  2$:     BIT     #DRY,@#DS1      ;RHDS1 SHOULD HAVE DPR,DRY
8732  043044  001004                          BNE     3$              ;BRANCH IF THERE
8733  043046  010337  001122                  MOV     R3,@#$BDADR     ;FAILING REGISTER RHDS1
8734  043052  104026                          ERROR   26              ;RHDS1 DOES NOT HAVE DPR,DRY
8735  043054  000407                          BR      4$              ;BRANCH OUT
8736  043056  032737  000400  001736  3$:     BIT     #DPR,@#DS1      ;RHDS1 SHOULD HAVE DPR,DRY
8737  043064  001004                          BNE     5$              ;BRANCH OUT AND CONTINUE IF THERE
8738  043066  010337  001122                  MOV     R3,@#$BDADR     ;FAILING REGISTER RHDS1
8739  043072  104026                          ERROR   26              ;RHDS1 DOES NOT HAVE DPR,DRY
8740
8741  043074  000207                  4$:     RTS     PC              ;RETURN TO TEST AND HALT
8742
8743  043076  062716  000006          5$:     ADD     #6,(SP)         ;ADJUST STACK TO GET OVER HALT IN TEST
8744  043102  000207                          RTS     PC              ;RETURN TO TEST AND CONTINUE TESTING
```

```
8745
8746
8747                                    ;*      WAIT LOOP
8748                                    ;*      ONE  LOOP OR ONE COUNT = 5.15 MICROSEC WITH BIPOLAR MEMORY (MIN)
8749                                    ;*      ONE LOOP OR ONE COUNT = 11.86 MICROSEC WITH CORE (MIN)
8750                                    ;*      WITH CORE ERROR IS INDICATED AFTER ABOUT 650 MILLISEC (MIN)
8751
8752  043104  177777                    TIMCNT: 177777                  ;WAITING COUNT
8753  043106  010046                    WAIT.T: MOV     R0,-(SP)                ;SAVE R0
8754  043110  016600  000002                    MOV     2(SP),R0                ;GET ADRESS OF REG. ADDRESS
8755  043114  010037  001204                    MOV     R0,@#$TMP3              ;WAT PC+2 IN $TMP3
8756  043120  162737  000002  001204            SUB     #2,@#$TMP3              ;WAT PC FOR TYPEOUT
8757  043126  012037  001176                    MOV     (R0)+,@#$TMP0           ;WAIT REGISTER ADDRESS
8758  043132  012037  001200                    MOV     (R0)+,@#$TMP1           ;WAIT ON BIT
8759  043136  010066  000002                    MOV     R0,2(SP)                ;RESTORE RETURN ON STACK
8760  043142  012600                            MOV     (SP)+,R0                ;RESTORE R0
8761  043144  013737  043104  001202            MOV     @#TIMCNT,@#$TMP2        ;TEMPORARY COUNT
8762  043152  033777  001200  136016  1$:       BIT     @#$TMP1,@$TMP0  ;IS REQUIRED BIT THERE?
8763  043160  001021                            BNE     2$              ;BRANCH IF YES
8764  043162  005337  001202                    DEC     @#$TMP2         ;COUNT
8765  043166  001371                            BNE     1$              ;BRANCH IF NOT TIME UP
8766  043170  013737  043104  001202            MOV     @#TIMCNT,@#$TMP2        ;TEMPORARY COUNT
8767  043176  033777  001200  135772  3$:       BIT     @#$TMP1,@$TMP0  ;IS REQUIRED BIT THERE?
8768  043204  001007                            BNE     2$              ;BRANCH IF YES
8769  043206  005337  001202                    DEC     @#$TMP2         ;COUNT
8770  043212  001371                            BNE     3$              ;BRANCH IF NOT TIME UP
8771  043214  017737  135756  001126            MOV     @$TMP0,@#$BDDAT ;REGISTER CONTENTS
8772  043222  104016                            ERROR   16              ;WAITED ON BIT FAILED TO SET
8773  043224  000002                    2$:     RTI
8774
8775
8776
8777                                    ;*      CALL FOR THE ABOVE WAITLOOP IS
8778                                    ;*
8779                                    ;*      MOV     @A,@#X$         ;A CONTAINS REGISTER ADDRESS
8780                                    ;*      -       -       -       ;HENCE X$ WILL HAVE ABSOLUTE REG. ADR.
8781                                    ;*      -       -       -
8782                                    ;*      -       -       -
8783                                    ;*      WAT
8784                                    ;*X$:   0                       ;ABSOLUTE REG. ADDRESS UNDER WAIT
8785                                    ;*      .WORD   0               ;BIT WAITED FOR
8786                                    ;*                              ;CONTINUE
8787
```

```
8788
8789                                    .SBTTL  SAVE ROUTINE
8790
8791                                    ;*THIS IS A SUBROUTINE TO SAVE REGISTERS
8792                                    ;*IN THE REGISTER TABLE TO ANY LOCATION
8793                                    ;*THE CALL IS
8794                                    ;*JSR   R0,@#SAVER
8795                                    ;*FROM
8796                                    ;*TO
8797                                    ;*NUMBER OF WORDS SAVED
8798
8799  043226                    SAVER:
8800  043226  010146                   MOV     R1,-(SP)        ;;PUSH R1 ON STACK
8801  043230  010246                   MOV     R2,-(SP)        ;;PUSH R2 ON STACK
8802  043232  010346                   MOV     R3,-(SP)        ;;PUSH R3 ON STACK
8803  043234  012001                   MOV     (R0)+,R1        ;FROM
8804  043236  012002                   MOV     (R0)+,R2        ;TO
8805  043240  012003                   MOV     (R0)+,R3        ;NUMBER
8806  043242  013122            1$:    MOV     @(R1)+,(R2)+    ;SAVE REGISTER CONTENTS
8807  043244  005303                   DEC     R3              ;COUNT
8808  043246  001375                   BNE     1$              ;BRANCH IF NOT DONE
8809  043250  012603                   MOV     (SP)+,R3        ;;POP STACK INTO R3
8810  043252  012602                   MOV     (SP)+,R2        ;;POP STACK INTO R2
8811  043254  012601                   MOV     (SP)+,R1        ;;POP STACK INTO R1
8812  043256  000200                   RTS     R0
8813
8814
8815
8816
8817                                    .SBTTL  WRITE CHECK ROUTINE
8818
8819                                    ;*THIS IS A SUBROUTINE TO DO WRITE CHECK HEADER AND DATA
8820                                    ;*CYLINDER 0, TRACK 1, SECTOR 1, KEYS 0
8821
8822                                    ;*THESE ARE TO SET UP FOR DISKLESS USE ONLY
8823  043260  012737  010000  047200  WRCHHD: MOV     #FMT22,@#CYL    ;CYLINDER 0 FORMAT 16 BIT WORDS
8824  043266  112737  000001  047203         MOVB    #1,@#SECOTR+1   ;TRACK=1
8825  043274  112737  000001  047202         MOVB    #1,@#SECOTR     ;SECTOR=1
8826  043302  005037  047204                 CLR     @#KEY1          ;KEY1=0
8827  043306  005037  047206                 CLR     @#KEY2          ;KEY2=0
8828  043312  012737  000044  047260         MOV     #36.,DAWORD     ;NO OF DATA WORDS
8829  043320  005037  047210                 CLR     @#X             ;THIS IS A READ OPERATION
8830  043324  004537  043742                 JSR     R5,@#CRC        ;GO TO CALCULATE CRC
8831  043330  047200                        CYL
8832  043332  051100                        WCRC
8833
8834                                    ;*THESE ARE REGULAR SETUPS
8835
8836  043334  004737  042534                 JSR     PC,@#CLDISK     ;SET UP GENERAL REGISTERS
8837                                                                   ;AND CLEAR DISK REGISTERS
8838  043340  012777  177730  136264         MOV     #-40.,@RHWC     ;36 DATA WORDS 4 HEADER WORDS
8839  043346  012777  003154  136260         MOV     #REINTO,@RHBA   ;STARTING ADDRESS OF READ BUFFER
8840  043354  112746  000001                 MOVB    #1,-(SP)        ;SECTOR=1
8841  043360  112766  000001  000001         MOVB    #1,1(SP)        ;TRACK=1 IN UPPER BYTE
8842  043366  012677  136252                 MOV     (SP)+,@RHDST    ;TRACK=1, SECTOR=1 IN RHDST
8843  043372  012777  014000  136250         MOV     #FMT22!ECI,@RHOF ;16 BIT WORDS
```

```
8844                                                        ;ECC CORRECTION INHIBIT BECAUSE
8845                                                        ;ECC LOGIC IS NOT CHECKED YET
8846  043400  005077  136246          CLR     @RHCA           ;CYLINDER=0
8847  043404  004737  042570          JSR     PC,@#CHECKT     ;CHECK THAT DVA,RDY,DPR,DRY = 1
8848  043410  104401  062145          TYPE    ,CPHALT         ;AND THAT NO OTHERS = 1.  CANNOT CON-
8849                                                        ;TINUE TESTING IF BOTH AREN'T TRUE
8850  043414  000000                  HALT                    ;STOP THE TEST
8851  043416  013711  002060          MOV     @#WRCHDT,@R1    ;WRITE CHECK HEADER AND DATA=52
8852                                                        ;INTO RRCS1
8853  043422  004737  047040          JSR     PC,@#COMHD      ;WRITE CHECK HEADER AND DATA
8854                                                        ;SAME AS READ HEADER AND DATA
8855
8856  043426  000207                  RTS     PC              ;RETURN TO WRITE CHECK TEST
8857
8858
```

```
8859
8860
8861                                    .SBTTL  COMPARE ROUTINE
8862
8863
8864                                    ;*THIS IS A SUBROUTINE TO COMPAPE TWO BLOCKS IN MEMORY
8865                                    ;*R1 HAS GOOD DATA BUFFER ADDRESS
8866                                    ;*R2 HAS TEST DATA BUFFER ADDRESS
8867                                    ;*$TMP0 HAS ADDRESS OF RETURN ON ERROR TO PRINT HEADER
8868                                    ;*$TMP1 HAS ADDRESS OF RETURN ON ERROR TO PRINT DATA
8869                                    ;*R3 HAS NUMBER OF WORDS TO BE COMPARED
8870                                    ;*R4 HAS ONE MORE THAN NUMBER OF WORDS TO BE COMPARED
8871
8872  043430                            COMPAR:
8873  043430  010146                           MOV     R1,-(SP)        ;;PUSH R1 ON STACK
8874  043432  010246                           MOV     R2,-(SP)        ;;PUSH R2 ON STACK
8875  043434  010346                           MOV     R3,-(SP)        ;;PUSH R3 ON STACK
8876  043436  010446                           MOV     R4,-(SP)        ;;PUSH R4 ON STACK
8877  043440  010546                           MOV     R5,-(SP)        ;;PUSH R5 ON STACK
8878  043442  012001                           MOV     (R0)+,R1        ;ADDRESS OF GOOD DATA BUFFER
8879  043444  012002                           MOV     (R0)+,R2        ;ADDRESS OF TEST DATA BUFFER
8880  043446  012003                           MOV     (R0)+,R3        ;NO OF WORDS TO BE COMPARED
8881  043450  012037  001176                  MOV     (R0)+,$TMP0     ;RETURN ON ERROR TO PRINT HEADER
8882  043454  012037  001200                  MOV     (R0)+,$TMP1     ;RETURN ON ERROR TO PRINT DATA
8883  043460  011000                           MOV     (R0),R0         ;RETURN ON NO ERROR
8884  043462  010304                           MOV     R3,R4           ;NO OF WORDS TO BE COMPARED
8885  043464  005204                           INC     R4
8886  043466  010437  047320          1$:     MOV     R4,@#ERWORD     ;FOR ERROR WORD NO
8887  043472  022122                           CMP     (R1)+,(R2)+     ;COMPARE GOOD WITH TEST DATA
8888  043474  001426                           BEQ     3$              ;BRANCH IF GOOD
8889
8890  043476  014137  001124                  MOV     -(R1),@#$GDDAT  ;GOOD DATA
8891  043502  014237  001126                  MOV     -(R2),@#$BDDAT  ;BAD DATA
8892  043506  160337  047320                  SUB     R3,@#ERWORD     ;ERROR WORD NO.
8893  043512  005737  002006                  TST     @#ERFLG$        ;ANY ERRORS ALREAY THERE
8894  043516  001003                           BNE     2$              ;BRANCH IF YES
8895  043520  004777  135452                  JSR     PC,@$TMP0       ;RETURN TO PRINT HEADER
8896  043524  000402                           BR      5$              ;BRANCH TO AVOID PRINTING NEXT ERROR
8897  043526  004777  135446          2$:     JSR     PC,@$TMP1       ;RETURN TO PRINT DATA
8898  043532  022122                  5$:     CMP     (R1)+,(R2)+     ;UNDO -(R1) AND -(R2) FOR ERRORS
8899  043534  017746  135400                  MOV     @SWR,-(SP)      ;GET SWITCH SETTING
8900  043540  042716  177177                  BIC     #^C600,(SP)     ;KEEP ONLY SWITCH 7 AND 8
8901  043544  022726  000200                  CMP     #SW07,(SP)+     ;IS 7 SET AND 8 RESET
8902  043550  001402                           BEQ     4$              ;BRANCH OUT IF YES
8903  043552  005303                  3$:     DEC     R3              ;COUNT
8904  043554  001344                           BNE     1$              ;BRANCH IF ALL NOT DEVICE
8905  043556                          4$:
8906  043556  012605                           MOV     (SP)+,R5        ;;POP STACK INTO R5
8907  043560  012604                           MOV     (SP)+,R4        ;;POP STACK INTO R4
8908  043562  012603                           MOV     (SP)+,R3        ;;POP STACK INTO R3
8909  043564  012602                           MOV     (SP)+,R2        ;;POP STACK INTO R2
8910  043566  012601                           MOV     (SP)+,R1        ;;POP STACK INTO R1
8911  043570  000200                           RTS     R0              ;RETURN TO MAIN PROGRAM
8912
8913
8914
```

```
8915
8916                                        ;*THIS IS A SUBROUTINE TO DO WRITE CHECK DATA
8917                                        ;*CYLINDER 0, TRACK 1, SECTOR 1, KEYS 0
8918
8919                                        ;*THESE ARE TO SET UP FOR DISKLESS USE ONLY
8920
8921  043572  012737  010000  047200  WRCHDA: MOV     #FMT22,@#CYL    ;CYLINDER 0 FORMAT 16 BIT WORDS
8922  043600  112737  000001  047203          MOVB    #1,@#SECOTR+1   ;TRACK=1
8923  043606  112737  000001  047202          MOVB    #1,@#SECOTR     ;SECTOR=1
8924  043614  005037  047204                  CLR     @#KEY1          ;KEY1=0
8925  043620  005037  047206                  CLR     @#KEY2          ;KEY2=0
8926  043624  012737  000040  047260          MOV     #32.,@#DAWORD   ;NO OF DATA WORDS
8927  043632  005037  047210                  CLR     @#X             ;THIS IS A READ OPERATION
8928
8929  043636  004537  043742                  JSR     R5,@#CRC        ;GO TO CALCULATE CRC
8930  043642  047200                          CYL
8931  043644  051100                          WCRC
8932
8933                                        ;*THESE ARE REGULAR SETUPS
8934
8935  043646  004737  042534                  JSR     PC,@#CLDISK     ;SET UP GENERAL REGISTERS
8936                                                                  ;AND CLEAR DISK REGISTERS
8937
8938  043652  012777  177740  135752          MOV     #-32.,@RHWC     ;36 DATA WORDS 4 HEADER WORDS
8939  043660  012777  003154  135746          MOV     #REINTO,@RHBA   ;STARTING ADDRESS OF READ BUFFER
8940  043666  112746  000001                  MOVB    #1,-(SP)        ;SECTOR=1
8941  043672  112766  000001  000001          MOVB    #1,1(SP)        ;TRACK=1 IN UPPER BYTE
8942  043700  012677  135740                  MOV     (SP)+,@RHDST    ;TRACK=1, SECTOR=1 IN RHDST
8943  043704  012777. 014000  135736          MOV     #FMT22!ECI,@RHOF        ;16 BIT WORDS
8944                                                                  ;ECC CORRECTION INHIBIT BECAUSE
8945                                                                  ;ECC LOGIC IS NOT CHECKED YET
8946  043712  005077  135734                  CLR     @RHCA           ;CYLINDER=0
8947  043716  004737  042570                  JSR     PC,@#CHECKT     ;CHECK THAT DVA,RDY,DPR,DRY = 1
8948  043722  104401  062145                  TYPE    ,CPHALT         ;AND THAT NO OTHERS = 1.  CANNOT CON-
8949                                                                  ;TINUE TESTING IF BOTH AREN'T TRUE
8950  043726  000000                          HALT                    ;STOP THE TEST
8951  043730  013711  002056                  MOV     @#WRCHEK,@R1    ;WRITE CHECK DATA=50 INTO RHCS1
8952  043734  004737  047040                  JSR     PC,@#COMHD      ;WRITE CHECK HEADER AND DATA
8953                                                                  ;SAME AS READ HEADER AND DATA
8954
8955  043740  000207                          RTS     PC              ;RETURN TO WRITE CHECK TEST
```

```
8956
8957                                    .SBTTL  CRC GENERATION ROUTINE
8958
8959
8960                                    ;*THIS IS A SUBROUTINE TO CALCULATE CRC FOR THE FOUR
8961                                    ;*HEADER WORDS AND STORE THEM IN "WCRC" AND "GCRC"
8962                                    ;*R1 - REGISTER FOR CRC, INCREMENTED CRC VALUE IS HERE
8963                                    ;*R2 - THIS HAS BIT POSITION 2 VALUE C
8964                                    ;*R3 - THIS HAS BIT POSITION 16 I.E. OUTPUT BIT VALUE B
8965                                    ;*R4 - THIS HAS BIT POSITION 15 VALUE E
8966                                    ;*$TMP0 - NUMBER OF WORDS
8967                                    ;*$TMP2 - NUMBER OF BITS PER WORD = 16
8968                                    ;*$TMP3 - TEMPORARY REG.
8969                                    ;*$TMP4 - TEMPORARY REG TO TRANSFER CARRY
8970                                    ;*$TMP5 - THIS HAS DATA BIT VALUE D
8971
8972                                    ;*FETCH DATA BIT D
8973                                    ;*B = D XOR 16
8974                                    ;*C = B XOR 2
8975                                    ;*E = B XOR 15
8976                                    ;*ROTATE RIGHT ONE POSITION
8977                                    ;*B GOES TO POSITION 1
8978                                    ;*C GOES TO POSITION 3
8979                                    ;*E GOES TO POSITION 16
8980                                    ;*REPET 64 TIMES
8981                                    ;*CALL  JSR     R5,@#CRC
8982                                    ;*X     ;FIRST LOCATION AT
8983                                    ;*Y     ;PUT CRC IN WCRC FOR READ GCRC FOR WRITE
8984
8985  043742                        CRC:
8986  043742  010046                    MOV     R0,-(SP)        ;;PUSH R0 ON STACK
8987  043744  012500                    MOV     (R5)+,R0        ;GET POINTER TO CYL NO.
8988  043746  010146                    MOV     R1,-(SP)        ;;PUSH R1 ON STACK
8989  043750  010246                    MOV     R2,-(SP)        ;;PUSH R2 ON STACK
8990  043752  010346                    MOV     R3,-(SP)        ;;PUSH R3 ON STACK
8991  043754  010446                    MOV     R4,-(SP)        ;;PUSH R4 ON STACK
8992  043756  005001                    CLR     R1              ;CLEAR WORKING LOCATION
8993  043760  005037  001210            CLR     @#$TMP5
8994  043764  012737  000004  001176    MOV     #4,@#$TMP0      ;WORD COUNT
8995  043772  012037  001204        16$:MOV     (R0)+,@#$TMP3   ;TEMPORARY WORD STORAGE
8996  043776  012737  000020  001202    MOV     #16.,$TMP2      ;BIT COUNT
8997  044004  013737  001204  001206    MOV     @#$TMP3,@#$TMP4 ;TEMPORARY WORD STORAGE
8998  044012  006037  001204        15$:ROR     @#$TMP3         ;GET LSB INTO "C"
8999  044016  006037  001210            ROR     @#$TMP5         ;GET ABOVE "C" INTO $TMP5
9000  044022  032701  000001            BIT     #BIT0,R1        ;IS POSITION 15 HIGH
9001  044026  001403                    BEQ     1$              ;BRANCH IF POSITION 16 LOW
9002  044030  012703  100000            MOV     #BIT15,R3       ;GET POSITION 16
9003  044034  000401                    BR      2$
9004  044036  005003                1$: CLR     R3              ;GET POSITION 16
9005  044040  063703  001210        2$: ADD     @#$TMP5,R3      ;XOR POSITION 16 WITH D
9006                                                            ;TO GIVE B
9007  044044  032701  040000            BIT     #BIT14,R1       ;IS POSITION 2 HIGH
9008  044050  001403                    BEQ     3$              ;BRANCH IF POSITION 2 LOW
9009  044052  012702  100000            MOV     #BIT15,R2       ;GET POSITION 2
9010  044056  000401                    BR      4$
9011  044060  005002                3$: CLR     R2              ;GET POSITION 2
```

```
9012  044062  060302                4$:     ADD     R3,R2           ;XOR B WITH POSITION 2
9013                                                                  ;TO GIVE C
9014  044064  032701  000002                BIT     #BIT1,R1        ;IS POSITION 15 HIGH
9015  044070  001403                        BEQ     5$              ;BRANCH IF POSITION 15 LOW
9016  044072  012704  100000                MOV     #BIT15,R4       ;GET POSITION 15
9017  044076  000401                        BR      6$
9018  044100  005004                5$:     CLR     R4              ;GET POSITION 15
9019  044102  060304                6$:     ADD     R3,R4           ;XOR POSITION 15 WITH B
9020                                                                  ;TO GIVE E
9021  044104  006037  001206                ROR     @#$TMP4         ;GET LSB INTO "C"
9022  044110  006001                        ROR     R1              ;GET ABOVE C INTO R1
9023  044112  005703                        TST     R3              ;TEST B
9024  044114  100403                        BMI     7$              ;BRANCH IF B=1
9025  044116  042701  100000                BIC     #BIT15,R1       ;SET B IN POSITION 1
9026  044122  000402                        BR      10$
9027  044124  052701  100000        7$:     BIS     #BIT15,R1       ;SET B IN POSITION 1
9028  044130  005702                10$:    TST     R2              ;TEST C
9029  044132  100403                        BMI     11$             ;BRANCH IF C=1
9030  044134  042701  020000                BIC     #BIT13,R1       ;GET C IN POSITION 3
9031  044140  000402                        BR      12$
9032  044142  052701  020000        11$:    BIS     #BIT13,R1       ;GET C IN POSITION 3
9033  044146  005704                12$:    TST     R4              ;TEST E
9034  044150  100403                        BMI     13$             ;BRANCH IF E=1
9035  044152  042701  000001                BIC     #BIT0,R1        ;GET E IN POSITION 16
9036  044156  000402                        BR      14$
9037  044160  052701  000001        13$:    BIS     #BIT0,R1        ;GET E IN POSITION 16
9038  044164  005337  001202        14$:    DEC     @#$TMP2         ;BIT COUNTER
9039  044170  001310                        BNE     15$             ;BRANCH IF 16 NOT DONE
9040  044172  005337  001176                DEC     @#$TMP0         ;WORD COUNTER
9041  044176  001275                        BNE     16$             ;BRANCH IF 4 NOT DONE
9042  044200  010135                        MOV     R1,@(R5)+       ;PUT CRC WHERE DESIRED
9043  044202  012604                        MOV     (SP)+,R4        ;;POP STACK INTO R4
9044  044204  012603                        MOV     (SP)+,R3        ;;POP STACK INTO R3
9045  044206  012602                        MOV     (SP)+,R2        ;;POP STACK INTO R2
9046  044210  012601                        MOV     (SP)+,R1        ;;POP STACK INTO R1
9047  044212  012600                        MOV     (SP)+,R0        ;;POP STACK INTO R0
9048  044214  000205                        RTS     R5
9049
9050
9051
9052
9053                                        ;*THIS IS A SUBROUTINE TO SET UP THE SIMULATOR DISK FOR
9054                                        ;*CYLINDER 0 (16 BITS PER WORD)
9055                                        ;*TRACK 1, SECTOR 1
9056                                        ;*KEY1 1
9057                                        ;*KEY2 1
9058                                        ;*CRC THROUGH THE JSR R5,@#CRC
9059                                        ;*256 WORDS OF 177400
9060
9061                                        ;*CALL  JSR     PC,@#SETDSK
9062
9063  044216                        SETDSK:
9064  044216  010046                        MOV     R0,-(SP)        ;;PUSH R0 ON STACK
9065  044220  010146                        MOV     R1,-(SP)        ;;PUSH R1 ON STACK
9066  044222  010246                        MOV     R2,-(SP)        ;;PUSH R2 ON STACK
9067  044224  012700  177400                MOV     #177400,R0      ;DATA IN THE DISK
```

```
9068  044230  012701  000400              MOV     #256.,R1        ;COUNTER
9069  044234  012702  051116              MOV     #DISK,R2        ;START OF SIMULATOR DISK
9070  044240  010022              1$:     MOV     R0,(R2)+        ;MOVE IN DATA
9071  044242  005301                      DEC     R1              ;COUNT FOR 256
9072  044244  001375                      BNE     1$              ;BRANCH IF 256 NOT COMPLETE
9073  044246  012701  000021              MOV     #17.,R1         ;2 ECC WORDS, 1 DATA GAP
9074                                                              ;14 TOLERANCE GAP
9075  044252  005022              2$:     CLR     (R2)+           ;CLEAR ECC,DATA GAP AND
9076                                                              ;TOLERANCE GAP
9077  044254  005301                      DEC     R1              ;COUNT
9078  044256  001375                      BNE     2$              ;BRANCH IF NOT COMPLETE
9079
9080                                      ;*NOW SET UP FOR DISKLESS USE
9081
9082  044260  012737  010000  047200      MOV     #FMT22,@#CYL    ;CYLINDER 0 (16 BIT WORDS)
9083  044266  112737  000001  047203      MOVB    #1,@#SECOTR+1   ;TRACK=1
9084  044274  112737  000001  047202      MOVB    #1,@#SECOTR     ;SECTOR=1
9085  044302  012737  000001  047204      MOV     #1,@#KEY1       ;KEY1=1
9086  044310  012737  000001  047206      MOV     #1,@#KEY2       ;KEY2=1
9087  044316  013737  000400  047260      MOV     256.,@#DAWORD   ;NO. OF DATA WORDS
9088  044324  004537  043742              JSR     R5,@#CRC        ;GO TO CALCULATE CRC
9089  044330  047200                      CYL                     ;FIRST CRC WORD
9090  044332  051100                      WCRC                    ;PUT CALCULATED CRC
9091  044334  012602                      MOV     (SP)+,R2        ;;POP STACK INTO R2
9092  044336  012601                      MOV     (SP)+,R1        ;;POP STACK INTO R1
9093  044340  012600                      MOV     (SP)+,R0        ;;POP STACK INTO R0
9094  044342  000207                      RTS     PC
```

```
 9095
 9096                                   ;*THIS IS A SUBROUTINE TO CHECK HEADER COMPARE ERROR
 9097                                   ;*(BIT #7) AND CRC ERROR (BIT #8)
 9098
 9099                                   ;*CALL JSR R0,@#HCCRCE
 9100
 9101                                   ;*      COM             ;COMMAND-READ HEADER AND DATA
 9102                                   ;                       ;       -WRITE DATA
 9103                                   ;*      C               ;CYLINDER
 9104                                   ;*      S               ;SECTOR
 9105                                   ;*      T               ;TRACK
 9106                                   ;*      -N.             ;WORD COUNT
 9107                                   ;*      B               ;RHBA BUFFER START
 9108                                   ;*      X               ;1=WRITE DATA 0=READ
 9109                                   ;*      H               ;H=1 HEADER CHECK, H=0 CRC CHECK
 9110
 9111  044344  010037  002014      HCCRCE: MOV     R0,@#PCJSR      ;SAVE PC OF JSR+4
 9112  044350  162737  000004  002014      SUB     #4,@#PCJSR      ;GET PC OF JSR
 9113  044356  004737  042534              JSR     PC,@#CLDISK     ;INIT AND SETUP GENERAL REG.
 9114  044362  004737  042570              JSR     PC,@#CHECKT     ;CHECK THAT DVA,RDY,DPR,DRY = 1
 9115  044366  104401  062145              TYPE    ,CPHALT         ;AND THAT NO OTHERS = 1.  CANNOT CON-
 9116                                                              ;TINUE TESTING IF BOTH AREN'T TRUE
 9117  044372  000000                      HALT                    ;STOP THE TEST
 9118  044374  011037  001210              MOV     (R0),@#$TMP5    ;SAVE COMMAND
 9119  044400  012011                      MOV     (R0)+,@R1       ;COMMAND
 9120  044402  012077  135244              MOV     (R0)+,@RHCA     ;CYLINDER
 9121  044406  112046                      MOVB    (R0)+,-(SP)     ;SECTOR
 9122  044410  105720                      TSTB    (R0)+           ;UP DATE R0
 9123  044412  112066  000001              MOVB    (R0)+,1(SP)     ;TRACK
 9124  044416  105720                      TSTB    (R0)+           ;UPDATE R0
 9125  044420  012677  135220              MOV     (SP)+,@RHDST    ;TRACK SECTOR
 9126  044424  012077  135202              MOV     (R0)+,@RHWC     ;NO. OF DATA WORDS +4 HEADER
 9127                                                              ;IF A READ HEADER AND DATA
 9128  044430  012077  135200              MOV     (R0)+,@RHBA     ;STARTING ADDRESS OF BUFFER
 9129  044434  012037  047210              MOV     (R0)+,@#X       ;X=0 READ HEADER AND DATA
 9130                                                              ;X=1 WRITE DATA
 9131  044440  012777  014000  135202      MOV     #FMT22!ECI,@RHOF ;16 BITS PER WORD
 9132                                                              ;ECC CORRECTION INHIBIT
 9133  044446  005037  002006              CLR     @#ERFLG$        ;CLEAR ERROR FLAG
 9134  044452  004737  047040              JSR     PC,@#COMHD      ;COMMAND
 9135
 9136                                   ;*IF THE PROGRAM COMES BACK HERE WITHOUT ERROR PRINTOUTS
 9137                                   ;*FROM THE "COMHD" ROUTINE THAT MEANS SECTOR GAP,
 9138                                   ;*FIRST SYNC, HEADER, HEADER CRC, HEADER GAP AND
 9139                                   ;*SYNC BYTE HAVE GONE BY AND SYNCS WERE CORRECTLY
 9140                                   ;*DETECTED
 9141                                   ;*HEADER AND DATA ARE TO BE CHECKED.
 9142
 9143  044456  004737  042140              JSR     PC,@#PUTREG     ;SAVE REGISTERS
 9144  044462  005737  002006              TST     @#ERFLG$        ;ANY ERRORS ALREADY THERE
 9145  044466  001034                      BNE     10$             ;BRANCH IF YES
 9146  044470  005737  047210              TST     @#X             ;IS THIS A READ
 9147  044474  001015                      BNE     3$              ;IF A WRITE DATA BRANCH
```

```
9148
9149                                    ;*NOW THE READ BUFFER WILL BE CHECKED
9150                                    ;*HEADER SHOULD BE COMPLETELY READ AS WRITTEN
9151                                    ;*NO DATA WORDS SHOULD BE READ
9152                                    ;*REINTO BUFFER HAS BEEN FILLED WITH 0
9153                                    ;*WRFROM BUFFER HAS BEEN FILLED WITH EXPECTED DATA
9154
9155  044476  004037  043430           JSR     R0,@#COMPAR     ;CHECK
9156  044502  002110                   WRFROM                  ;GOOD DATA
9157  044504  003154                   REINTO                  ;TEST BUFFER
9158  044506  000400                   256.                    ;4 HEADER 252 DATA
9159  044510  044516                   1$                      ;RETURN POINT FOR ERROR HEADER
9160  044512  044522                   2$                      ;RETURN POINT FOR ERROR DATA
9161  044514  044560                   10$                     ;RETURN FOR GOOD COMPARISON
9162  044516  104004           1$:     ERROR   4               ;READ NEXT ERROR 5
9163  044520  000207                   RTS     PC              ;RETURN TO COMPARISON SUBROUTINE
9164  044522  104005           2$:     ERROR   5               ;WORD NO 1 THRU 4 ARE
9165                                                           ;HEADER WORDS AND HENCE
9166                                                           ;SHOULD BE READ AS WRITTEN ON
9167                                                           ;DISK, WORD NOS. 5 ONWARDS
9168                                                           ;SHOULD NOT BE READ AND HENCE
9169                                                           ;READ INTO BUFFER
9170                                                           ;SHOULD BE UNCHANGED
9171  044524  000207                   RTS     PC              ;RETURN TO COMPARISON
9172
9173  044526  000414                   BR      10$             ;JUMP OUT
9174
9175                                    ;*NOW THE DISK WILL BE CHECKED
9176                                    ;*NO DATA SHOULD BE WRITTEN
9177                                    ;*REINTO BUFFER HAS BEEN FILLED WITH EXPECTED DATA
9178                                    ;*DISK HAS BEEN FILLED WITH 177400
9179                                    ;*WRFROM HAS BEEN FILLED WITH 125252
9180
9181  044530  004037  043430   3$:     JSR     R0,@#COMPAR     ;CHECK
9182  044534  003154                   REINTO                  ;GOOD DATA BUFFER
9183  044536  051116                   DISK                    ;TEST BUFFER
9184  044540  000400                   256.
9185  044542  044550                   4$                      ;RETURN POINT FOR ERROR HEADER
9186  044544  044554                   5$                      ;RETURN POINT FOR ERROR DATA
9187  044546  044560                   10$                     ;RETURN POINT FOR GOOD COMPARISON
9188  044550  104004           4$:     ERROR   4               ;READ NEXT ERROR 5
9189  044552  000207                   RTS     PC              ;RETURN TO COMPARISON SUBROUTINE
9190  044554  104005           5$:     ERROR   5               ;WORD NO ARE ALL DATA
9191                                                           ;WORDS THE SHOULD NOT
9192                                                           ;HAVE BEEN CHANGED BY THE
9193                                                           ;WRITE COMMAND
9194  044556  000207                   RTS     PC              ;RETURN TO COMPARISON SUBROUTINE
9195  044560  005720           10$:    TST     (R0)+           ;IS THIS A HCRC OR HCE CHECK?
9196  044562  001442                   BEQ     6$              ;BRANCH IF HCRC
9197  044564  022737  000072  001210   CMP     #72,@#$TMP5     ;IS THIS A READ COMMAND
9198  044572  001417                   BEQ     11$             ;BRANCH IF YES
9199  044574  017737  135042  001126   MOV     @RHER1,@#$BDDAT ;TEST DATA
9200  044602  022737  000200  001126   CMP     #HCE,@#$BDDAT   ;ONLY HEADER COMPARE BIT?
9201                                                           ;SHOULD BE SET
9202  044610  001470                   BEQ     7$              ;BRANCH IF GOOD
9203  044612  013737  001642  042204   MOV     @#RHER1,@#REGADR ;REGISTER ADDRESS RHER1
```

MI

```
9204  044620  012737  000200  001124          MOV     #HCE,@#$GDDAT   ;GOOD DATA
9205  044626  104027                          ERROR   27              ;AFTER AN ERROR ON THE
9206                                                                  ;HEADER ONLY HCE SHOULD
9207  044630  000460                          BR      7$              ;BE SET
9208  044632                          11$:
9209  044634  017737  175004  001126          MOV     @RHER1,@#$BDDAT ;TEST DATA
9210  044640  022737  100200  001126          CMP     #DCK!HCE,@#$BDDAT ;ONLY HEADER COMPARE BIT?
9211                                                                  ;SHOULD BE SET
9212                                                                  ;DCK IS SET BECAUSE ECC IS NOT READ
9213  044646  001451                          BEQ     7$              ;BRANCH IF GOOD
9214  044650  013737  001642  042204          MOV     @#RHER1,@#REGADR ;REGISTER ADDRESS RHER1
9215  044656  012737  100200  001124          MOV     #DCK!HCE,@#$GDDAT       ;GOOD DATA
9216  044664  104027                          ERROR   27              ;AFTER AN ERROR ON THE
9217                                                                  ;HEADER ONLY HCE SHOULD
9218  044666  000441                          BR      7$              ;BE SET
9219  044670  022737  000072  001210  6$:     CMP     #72,@#$TMP5     ;IS THIS A READ COMMAND?
9220  044676  001417                          BEQ     12$             ;BRANCH IF A READ
9221  044700  017737  134736  001126          MOV     @RHER1,@#$BDDAT ;TEST DATA
9222  044706  022737  000400  001126          CMP     #HCRC,@#$BDDAT  ;ONLY CRC ERROR SHOULD BE THERE
9223  044714  001426                          BEQ     7$
9224  044716  013737  001642  042204          MOV     @#RHER1,@#REGADR ;REG. ADDR = RHER1
9225  044724  012737  000400  001124          MOV     #HCRC,@#$GDDAT  ;GOOD DATA
9226  044732  104027                          ERROR   27              ;AFTER A CRC ERROR ONLY CRC
9227                                                                  ;SHOULD BE SET
9228  044734  000416                          BR      7$              ;BRANCH OUT
9229  044736  017737  134700  001126  12$:    MOV     @RHER1,@#$BDDAT ;TEST DATA
9230
9231  044744  022737  100400  001126          CMP     #DCK!HCRC,@#$BDDAT;HCRC AND DCK SHOULD BE SET
9232                                                                  ;DCK IS SET BECAUSE ECC IS NOT READ
9233  044752  001407                          BEQ     7$              ;BRANCH IF GOOD
9234  044754  012737  100400  001124          MOV     #DCK!HCRC,@#$GDDAT;GOOD DATA
9235  044762  013737  001642  042204          MOV     @#RHER1,@#REGADR;FAILING REGISTER RHER1
9236  044770  104027                          ERROR   27              ;AFTER A CRC ERROR ON A READ
9237                                                                  ;DCK AND HCRC SHOULD BE SET
9238                                                                  ;DCK IS SET BECAUSE ECC IS NOT READ
9239  044772  000200                  7$:     RTS     R0              ;RETURN TO MAIN TEST
9240
9241
9242
9243                                          ;*THIS IS A SUBROUTINE TO LEAVE AT THE MIDDLE OF
9244                                          ;*A WRITE HEADER AND DATA COMMAND
9245                                          ;*IT TRYS TO GET SECTOR 10, TRACK 0, CYLINDER 0
9246                                          ;*BUT COMES OUT AFTER ONE SECTOR
9247                                          ;*THE COMMAND OS JSR PC,@#MIDDLE
9248                                          ;*BAI IS SET
9249
9250  044774                          MIDDLE:
9251  044774  010046                          MOV     R0,-(SP)        ;;PUSH R0 ON STACK
9252  044776  010146                          MOV     R1,-(SP)        ;;PUSH R1 ON STACK
9253  045000  013777  002064  134632          MOV     @#WRIFOR,@RHCS1 ;WRITE HEADER AND DATA=62
9254                                                                  ;IN RHCS1
9255  045006  012777  177766  134616          MOV     #-10.,@RHWC     ;10 WORDS
9256  045014  012777  002110  134612          MOV     #WRFROM,@RHBA   ;BUS ADDRESS=WRFROM
9257  045022  012777  000010  134614          MOV     #10,@RHDST      ;DESIRED TRACK=0 SECTOR=10
9258  045030  052777  000010  134600          BIS     #BAI,@RHCS2     ;BUS ADDRESS INCREMENT INHIBIT
9259  045036  012777  010000  134604          MOV     #FMT22,@RHOF    ;FORMAT 16 BIT WORDS
```

A2

```
9260  045044  005077  134602                  CLR     @RHCA           ;CYLINDER=0
9261  045050  012737  000001  045076          MOV     #1,@#MID        ;SECTOR IS SET TO 1 SO THAT
9262                                                                    ;WE CAN GET OUT AT THE
9263                                                                    ;MIDDLE OF AN OPERATION
9264                                                                    ;LOOKING FOR SECTOR 10
9265  045056  012777  000001  134574          MOV     #DMD,@RHMR      ;SET DIAGNOSTIC MODE
9266  045064  052777  000001  134546          BIS     #GO,@RHCS1              ;GO TO RHCS1 WITH 62
9267  045072  004137  053246                  JSR     R1,@#SEARCH
9268  045076  000000                  MID:    .WORD   0               ;SECTOR
9269  045100  012601                          MOV     (SP)+,R1        ;;POP STACK INTO R1
9270  045102  012600                          MOV     (SP)+,R0        ;;POP STACK INTO R0
9271  045104  000207                          RTS     PC
9272
9273
9274
9275                                          .SBTTL  JAM CURRENT CYLINDER ROUTINE
9276
9277
9278                                  ;*THIS SUBROUTINE WILL CHANGE THE CURRENT CYLINDER REGISTER
9279                                  ;*THIS IS DONE BY GIVING A SEEK COMMAND THEN AN INIT
9280                                  ;*WHICH WILL LOAD THE CURRENT CYLINDER WITH THE DESIRED CYLINDER VALUE
9281                                  ;*
9282                                  ;*CALL IS
9283                                  ;*JSR   R0,@#MAKECYL
9284                                  ;*XC            ;DESIRED VALUE OF CURRENT CYLINDER
9285
9286
9287  045106                  MAKECYL:
9288  045106  010546                          MOV     R5,-(SP)        ;;PUSH R5 ON STACK
9289  045110  010037  002014                  MOV     R0,@#PCJSP      ;PC OF JSR+4
9290  045114  162737  000004  002014          SUB     #4,@#PCJSR      ;SAVE PC OF JSR
9291  045122  012005                          MOV     (R0)+,R5        ;GETTING READY TO FILL DESIRED CYLINDER
9292  045124  010577  134522                  MOV     R5,@RHCA        ;FILL DESIRED CYLINDER REGISTER
9293  045130  005077  134510                  CLR     @RHDST          ;MAKE SURE DESIRED SECTOR TRACK IS NOT ILLEGAL
9294  045134  013777  002072  134476          MOV     @#SEECOM,@RHCS1 ;FILL SEEK COMMAND
9295  045142  012777  000001  134510          MOV     #DMD,@RHMR      ;SET DIAGNOSTIC MODE
9296  045150  052777  000001  134462          BIS     #GO,@RHCS1      ;GO TO SEEK
9297  045156  000240                          NOP                     ;ALLOW TIME FOR SEEK TO HANG UP
9298  045160  000240                          NOP                     ;ALLOW TIME FOR SEEK TO HANG UP
9299  045162  000240                          NOP                     ;ALLOW TIME FOR SEEK TO HANG UP
9300  045164  000240                          NOP                     ;ALLOW TIME FOR SEEK TO HANG UP
9301  045166  004737  042534                  JSR     PC,@#CLDISK     ;GIVE INIT
9302  045172  017737  134500  001126          MOV     @RHCC,@#$BDDAT  ;TEST DATA
9303  045200  020537  001126                  CMP     R5,@#$BDDAT     ;COMPARE CURRENT CYLINDER
9304  045204  001406                          BEQ     1$              ;BRANCH IF GOOD
9305  045206  010537  001124                  MOV     R5,@#$GDDAT     ;GOOD VALUE OF RHCC
9306  045212  013737  001676  042204          MOV     @#RHCC,@#REGADR ;FAILING REGISTER ADDRESS
9307  045220  104030                          ERROR   30              ;CURRENT CYLINDER DOES NOT MATCH DESIRED CYLINDER
9308                                                                  ;REGISTER AFTER A SEEK AND AN INIT
9309  045222                          1$:
9310  045222  012605                          MOV     (SP)+,R5        ;;POP STACK INTO R5
9311  045224  000200                          RTS     R0
9312
9313
9314                                          .SBTTL  ECC GENERATION AND COMPARISON ROUTINE
9315
```

```
9316
9317
9318
9319                                    ;*THIS SUBROUTINE GENERATES AND TESTS ECC
9320                                    ;*CALL    JSR PC,ECTEST
9321
9322
9323
9324
9325
9326          100000                    PIE1    =100000
9327          040000                    PIE2    =40000
9328          020000                    PIE3    =20000
9329          010000                    PIE4    =10000
9330          004000                    PIE5    =4000
9331          002000                    PIE6    =2000
9332          001000                    PIE7    =1000
9333          000400                    PIE8    =400
9334          000200                    PIE9    =200
9335          000100                    PIE10   =100
9336          000040                    PIE11   =40
9337          000020                    PIE12   =20
9338          000010                    PIE13   =10
9339          000004                    PIE14   =4
9340          000002                    PIE15   =2
9341          000001                    PIE16   =1
9342          100000                    PIE17   =100000
9343          040000                    PIE18   =40000
9344          020000                    PIE19   =20000
9345          010000                    PIE20   =10000
9346          004000                    PIE21   =4000
9347          002000                    PIE22   =2000
9348          001000                    PIE23   =1000
9349          000400                    PIE24   =400
9350          000200                    PIE25   =200
9351          000100                    PIE26   =100
9352          000040                    PIE27   =40
9353          000020                    PIE28   =20
9354          000010                    PIE29   =10
9355          000004                    PIE30   =4
9356          000002                    PIE31   =2
9357          000001                    PIE32   =1
9358
9359  045226  000000                    ECDATA: 0               ;DATA BIT FOR ECC
9360                                                            ;IF ALL ONES THEN CURRENT BIT IS A ONE
9361                                                            ;IF ZERO THEN CURRENT BIT IS A ZERO
9362
9363  045230  000000                    GECC1:  0               ;LOW ORDER ECC WORD TO BE GENERATED HERE
9364                                                            ;=R1
9365
9366  045232  000000                    GECC2:  0               ;HIGH ORDER ECC WORD TO BE GENERATED HERE
9367                                                            ;=R2
9368
9369  045234  000000                    TSECCG: 0               ;IF =177777 GENERATE AND TEST ECC FOR THIS BIT
9370                                                            ;IF =0 DO NOT GENERATE AND TEST ECC FOR THIS BIT
9371
```

```
9372  045236  113713                  NCODE:  38859.          ;N-CODE WORD
9373  045240  000000                  NCOUNT: 0               ;TEMPORARY N CODE
9374  045242  000000                  POSITI: 0               ;POSITION REGISTER
9375  045244  010041                  HARDER: 4129.           ;HARD ERROR COUNT
9376                                                          ;TRUE COUNT IS 4128 BUT AS COMPARES ARE
9377                                                          ;DONE ONE STAGE LATER SO 4129
9378  045246  000000                  DATENV: 0               ;DATA ENVELOPE FOR TYPE OUT
9379                                                          ;MAX FOR WRITE IS 4096
9380                                                          ;MAX FOR READ IS 4128
9381  045250  000000                  ZCODE:  0               ;LEADING ZEROS ENVELOPE FOR TYPE OUT
9382                                                          ;THIS IS SHUT OFF WHEN POSITION COUNTER
9383                                                          ;IN ENABLED
9384                                                          ;MAX COUNT IS 38859
9385
9386
9387
9388  045252  000000                  HADTMP: 0               ;TEMPORARY HARD ERROR COUNT
9389  045254  000000                  P3:     0
9390  045256  000000                  P12:    0
9391  045260  000000                  P22:    0
9392  045262  000000                  P24:    0
9393
9394
9395
9396
9397
9398  045264                          ECTEST:
9399  045264  010046                          MOV     R0,-(SP)        ;;PUSH R0 ON STACK
9400  045266  010146                          MOV     R1,-(SP)        ;;PUSH R1 ON STACK
9401  045270  010246                          MOV     R2,-(SP)        ;;PUSH R2 ON STACK
9402  045272  010346                          MOV     R3,-(SP)        ;;PUSH R3 ON STACK
9403  045274  010446                          MOV     R4,-(SP)        ;;PUSH R4 ON STACK
9404  045276  010546                          MOV     R5,-(SP)        ;;PUSH R5 ON STACK
9405  045300  013701  045230                  MOV     @#GECC1,R1              ;ECC1 WORD
9406  045304  013702  045232                  MOV     @#GECC2,R2              ;ECC2 WORD
9407  045310  005737  045226                  TST     @#ECDATA                ;IS CURRENT BIT A ONE
9408  045314  001406                          BEQ     2$                      ;BRANCH IF CURRENT DATA D=0
9409
9410                                          ;*IF CARRY IS NOT ZERO THEN D=1
9411                                          ;*INVERT X32 TO GIVE R0
9412
9413  045316  010103                  1$:     MOV     R1,R3
9414  045320  052703  177776                BIS     #^CPIE32,R3
9415  045324  005103                          COM     R3
9416  045326  010300                          MOV     R3,R0
9417  045330  000404                          BR      3$
9418
9419                                          ;*IF CARRY IS ZERO THEN D=0
9420                                          ;*X32 BECOMES R0
9421  045332  010103                  2$:     MOV     R1,R3
9422  045334  042703  177776                BIC     #^CPIE32,R3
9423  045340  010300                          MOV     R3,R0
9424
9425  045342  000241                  3$:     CLC
9426  045344  006000                          ROR     R0
9427  045346  006000                          ROR     R0
```

02

```
 9428  045350  005700                          TST     R0
 9429  045352  001462                          BEQ     10$             ;BRANCH IF R0=0
 9430                                          ;*INVERT X2
 9431
 9432  045354  010203                          MOV     R2,R3
 9433  045356  052703  137777                  BIS     #^CPIE2,R3
 9434  045362  005103                          COM     R3
 9435  045364  010337  045254                  MOV     R3,@#P3
 9436  045370  006237  045254                  ASR     @#P3
 9437
 9438                                          ;*INVERT X11
 9439
 9440
 9441  045374  010203                          MOV     R2,R3
 9442  045376  052703  177737                  BIS     #^CPIE11,R3
 9443  045402  005103                          COM     R3
 9444  045404  010337  045256                  MOV     R3,@#P12
 9445  045410  006237  045256                  ASR     @#P12
 9446
 9447                                          ;*INVERT X21
 9448
 9449  045414  010103                          MOV     R1,R3
 9450  045416  052703  173777                  BIS     #^CPIE21,R3
 9451  045422  005103                          COM     R3
 9452  045424  010337  045260                  MOV     R3,@#P22
 9453  045430  006237  045260                  ASR     @#P22
 9454
 9455                                          ;*INVERT X23
 9456
 9457  045434  010103                          MOV     R1,R3
 9458  045436  052703  176777                  BIS     #^CPIE23,R3
 9459  045442  005103                          COM     R3
 9460  045444  010337  045262                  MOV     R3,@#P24
 9461  045450  006237  045262                  ASR     @#P24
 9462
 9463                                          ;*NOW THAT R0 FOR POSITION 1
 9464                                          ;*        P3 FOR POSITION 3
 9465                                          ;*       P12 FOR POSITION 12
 9466                                          ;*       P22 FOR POSITION 22
 9467                                          ;*       P24 FOR POSITION 24
 9468                                          ;*ARE KNOWN THE ROTATE WILL BE DONE AND
 9469                                          ;*THESE BITS JAMED IN
 9470
 9471  045454  006002                          ROR     R2
 9472  045456  006001                          ROR     R1
 9473  045460  053700  045254                  BIS     @#P3,R0
 9474  045464  053700  045256                  BIS     @#P12,R0
 9475  045470  042702  120020                  BIC     #PIE1!PIE3!PIE12,R2
 9476  045474  050002                          BIS     R0,R2
 9477
 9478  045476  005000                          CLR     R0
 9479  045500  053700  045260                  BIS     @#P22,R0
 9480  045504  053700  045262                  BIS     @#P24,R0
 9481  045510  042701  002400                  BIC     #PIE22!PIE24,R1
 9482  045514  050001                          BIS     R0,R1
 9483  045516  000404                          BR      12$
```

```
9484
9485                                        ;*THE PROGRAM COMES HERE IF R0=0
9486                                        ;*SO AFTER ROTATE R0 GETS PUT INTO POSITION 1
9487
9488  045520  006002              10$:     ROR     R2
9489  045522  006001                       ROR     R1
9490  045524  042702  100000             BIC     #PIE1,R2
9491  045530  010137  045230     12$:     MOV     R1,@#GECC1      ;SAVE ECC1
9492  045534  010237  045232              MOV     R2,@#GECC2      ;SAVE ECC2
9493  045540  005737  045234              TST     @#TSECCG        ;IS HARDWARE TO BE CHECKED
9494                                                              ;IF =1777777 TEST HARDWARE
9495                                                              ;IF = 0 DO NOT TEST HARDWARE
9496  045544  001432                       BEQ     14$             ;BRANCH IF HARDWARE NOT TO BE CHECKED
9497
9498
9499                              ;*CHECK HARDWARE
9500  045546  032777  000400  133364       BIT     #SW8,@SWR       ;IS SWITCH 8 SET
9501  045554  001005                       BNE     15$             ;BRANCH IF SW8 IS SET
9502  045556  032777  000100  133354       BIT     #SW6,@SWR       ;IS SWITCH 6 SET
9503  045564  001401                       BEQ     15$             ;BRANCH IF SW6 IS NOT SET
9504  045566  000421                       BR      14$             ;IF SWITCH 8 IS NOT SET AND
9505                                                              ;SWITCH 6 IS SET THEN
9506                                                              ;DO NOT DO COMPARES
9507  045570  010146              15$:     MOV     R1,-(SP)        ;GOOD PATTERN REGISTER
9508  045572  042716  174000             BIC     #174000,(SP)    ;GET ONLY PATTERN BITS
9509  045576  022677  134070             CMP     (SP)+,@RHEC2    ;COMPARE PATTERN REGISTER
9510  045602  001404                       BEQ     13$             ;BRANCH IF GOOD
9511                                      ;*TO SAVE TIME
9512  045604  004737  042140             JSR     PC,@#PUTRFG     ;SAVE REGISTERS
9513  045610  104035                       ERROR   35              ;PATTERN REGISTER IN 11 BITS IN ERROR
9514  045612  000407                       BR      14$             ;BRANCH OUT
9515  045614  023777  045242  134046 13$:  CMP     @#POSITI,@RHEC1 ;COMPARE POISTION REGISTER
9516  045622  001403                       BEQ     14$             ;BRANCH IF GOOD
9517                                      ;*TO SAVE TIME
9518  045624  004737  042140             JSR     PC,@#PUTREG     ;SAVE REGISTERS
9519  045630  104035                       ERROR   35              ;POSITION REGISTER IN ERROR
9520                                                              ;"DATA ENVLOP" GIVES NUMBER OF CLOCK
9521                                                              ;PULSES FROM BEGINING OF COMMAND
9522                                                              ;THAT IS THE CLOCKS IN THE R/W DATA FIELD ENVELOPE
9523                                                              ;
9524                                                              ;IN A WRITE THERE ARE 10000 OCTAL CLOCKS
9525                                                              ;IN A READ THERE ARE 10040 OCTAL CLOCKS
9526                                                              ;
9527                                                              ;
9528                                                              ;"N-CODE ZEROS" GIVE THE NUMBER OF CLOCKS
9529                                                              ; GIVEN FOR THE LEADING ZEROS FIELD
9530                                                              ;MAX COUNT IS 113713 OCTAL
9531                                                              ;
9532                                                              ;"GOOD POSITION" GIVES NUMBER OF CLOCKS
9533                                                              ;GIVEN AFTER LEADING ZEROS WHICH IS FOR THE DATA
9534                                                              ;FIELD
9535                                                              ;MAX COUNT IS 10040 OR 10041 OCTAL
9536
9537
9538  045632              14$:
9539  045632  012605                       MOV     (SP)+,R5        ;;POP STACK INTO R5
```

F2

```
 9540  045634  012604                    MOV     (SP)+,R4        ;;POP STACK INTO R4
 9541  045636  012603                    MOV     (SP)+,R3        ;;POP STACK INTO R3
 9542  045640  012602                    MOV     (SP)+,R2        ;;POP STACK INTO R2
 9543  045642  012601                    MOV     (SP)+,R1        ;;POP STACK INTO R1
 9544  045644  012600                    MOV     (SP)+,R0        ;;POP STACK INTO R0
 9545  045646  000207                    RTS     PC
```

```
9546
9547
9548                                    ;*THIS SUBROUTINE WILL CONTROL THE ECC GENERATION ROUTINE
9549                                    ;*FOR ERROR CORRECTION PROCESS
9550                                    ;*CALL  JSR,    PC,@#ECORR
9551                                    ;*      XP      ;EXPECTED POSITION REGISTER WHEN CORRECTION IS COMPLETE
9552
9553
9554
9555  045650  000000                    ERPOS:  0                       ;POSITION REG. WHEN CORRECTION IS COMPLETE
9556
9557
9558
9559  045652  010037  002014            ECORR:  MOV     R0,@#PCJSR      ;SAVE PC OF JSR + 4
9560  045656  162737  000004  002014            SUB     #4,@#PCJSR      ;SAVE PC OF JSR
9561  045664  012037  045650                    MOV     (P0)+,@#ERPOS   ;GET POSITION REG. WHEN CORRECTION IS COMPLETE
9562  045670  010146                        MOV     R1,-(SP)        ;;PUSH R1 ON STACK
9563  045672  013701  001660                    MOV     @#RHMR,R1       ;MAINTENANCE REGISTER
9564  045676  012711  000001                    MOV     #DMD,@R1        ;SET DIAGNOSTIC MODE BIT
9565  045702  005037  045226                    CLR     @#ECDATA        ;ECC DATA IS ZERO
9566
9567
9568
9569  045706  005737  045242            1$:     TST     @#POSITI        ;IS  SOFTWARE POSITION NON ZERO
9570  045712  001007                        BNE     2$              ;BRANCH IF N-CODE S COMPLETE
9571  045714  005337  045240                    DEC     @#NCOUNT        ;DECREMENT N-CODE
9572  045720  001001                        BNE     6$              ;BRANCH IF N-CODE IS NOT COMPLETE
9573  045722  000403                        BR      2$              ;BRANCH  AS N-CODE IS COMPLETE
9574  045724  005237  045250            6$:     INC     @#ZCODE         ;INCREMENT CLOCKS GIVEN FOR LEADING ZEROS
9575  045730  000420                        BR      3$              ;BRANCH AS N-CODE IS NOT COMPLETE
9576                                                                    ;GO TO GIVE CLOCK AND TEST ECC
9577  045732  005237  045242            2$:     INC     @#POSITI        ;INCREMENT SOFTWARE POSITION
9578  045736  023737  045650  045242            CMP     @#ERPOS,@#POSITI;HAVE ENOUGH CLOCKS BEEN GIVEN TO DETECT ERROR
9579  045744  103012                        BHIS    3$              ;BRANCH IF MORE CLOCKS TO BE GIVEN
9580  045746  023737  045252  045242            CMP     @#HADTMP,@#POSITI;HAVE ENOUGH CLOCKS BEEN GIVEN FOR HARD ERROR
9581                                                                    ;THAT IS HAVE 4128 MORE CLOCKS BEEN GIVEN
9582  045754  001415                        BEQ     5$              ;BRANCH IF YES
9583  045756  032711  000400                    BIT     #ZER,@R1        ;CHECK ZERO DETECT BIT IN RHMR
9584  045762  001016                        BNE     4$              ;BRANCH IS ZER SET
9585                                        ;*TO SAVE TIME
9586  045764  004737  042140                    JSR     PC,@#PUTREG     ;SAVE REGISTERS
9587  045770  104034                        ERROR   34              ;ZERO DETECT BIT NOT HIGH
9588                                                                    ;WHEN 21 BITS IN ECC 32 BIT REGISTER IS  0
9589
9590
9591  045772  052711  000002            3$:     BIS     #MCLK,@R1       ;SET CLOCK
9592  045776  042711  000002                    BIC     #MCLK,@R1       ;CLEAR CLOCK
9593  046002  004737  045264                    JSR     PC,@#ECTEST     ;GO TO GENERATE AND TEST ECC
9594  046006  000737                        BR      1$              ;CONTINUE
9595
9596                                    ;*THIS EXTRA CLOCK IS TO BRING ECH HIGH
9597                                    ;*AFTER THIS CLOCK POSITION REGISTER MAY BE 10040 OR 10041 OCTAL
9598
9599  046010  052711  000002            5$:     BIS     #MCLK,@R1       ;SET CLOCK
9600  046014  042711  000002                    BIC     #MCLK,@R1       ;CLEAR CLOCK
9601
```

```
 9602  046020                          4$:
 9603  046020  012601                          MOV     (SP)+,R1        ;;POP STACK INTO R1
 9604  046022  000200                          RTS     R0
 9605
 9606
 9607
 9608
 9609
 9610
 9611                                  ;*THIS SUBROUTINE GENERATES THE ECC FOR WHAT IS ON DISK AND INSERTS THEM
 9612                                  ;*ON LOCATIONS "DISK+1000" AND "DISK+1002"
 9613
 9614
 9615
 9616  046024                          FILLEC:
 9617  046024  010046                          MOV     R0,-(SP)        ;;PUSH R0 ON STACK
 9618  046026  010146                          MOV     R1,-(SP)        ;;PUSH R1 ON STACK
 9619  046030  010246                          MOV     R2,-(SP)        ;;PUSH R2 ON STACK
 9620  046032  010346                          MOV     R3,-(SP)        ;;PUSH R3 ON STACK
 9621  046034  010446                          MOV     R4,-(SP)        ;;PUSH R4 ON STACK
 9622  046036  010546                          MOV     R5,-(SP)        ;;PUSH R5 ON STACK
 9623  046040  005037  045242                  CLR     @#POSITI        ;CLEAR POSITION
 9624  046044  005037  045230                  CLR     @#GECC1         ;CLEAR GECC1
 9625  046050  005037  045232                  CLR     @#GECC2         ;CLEAR
 9626  046054  012701  051116                  MOV     #DISK,R1        ;POINTER TO DATA FOR ECC GENERATION
 9627  046060  012702  000400                  MOV     #256.,R2        ;COUNTER FOR NUMBER OF DATA WORDS
 9628  046064  012703  000020          9$:     MOV     #16.,R3         ;COUNTER FOR NUMBER OF BITS PER WORD
 9629  046070  012104                          MOV     (R1)+,R4        ;DATA IN R4
 9630  046072  006004                  10$:    ROR     R4              ;GET ONE DATA BIT IN CARRY
 9631  046074  103004                          BCC     11$             ;BRANCH IF DATA BIT IS ZERO
 9632  046076  012737  177777  045226          MOV     #-1,@#ECDATA    ;ECC DATA BIT IS A ONE
 9633  046104  000402                          BP      12$             ;BRANCH TO GENERATE ECC
 9634  046106  005037  045226          11$:    CLR     @#ECDATA        ;ECC DATA BIT IS A ZERO
 9635  046112  004737  045264          12$:    JSR     PC,@#ECTEST     ;GO TO GENERATE ECC
 9636  046116  005303                          DEC     R3              ;DECREMENT BIT COUNT
 9637  046120  001364                          BNE     10$             ;BRANCH IF 16 BITS NOT DONE
 9638  046122  005302                          DEC     R2              ;DECREMENT WORD COUNT
 9639  046124  001357                          BNE     9$              ;BRANCH IF 256 WORDS NOT DONE
 9640  046126  013737  045230  052116          MOV     @#GECC1,@#DISK+<256.*2>;INSERT ECC1 ON DISK
 9641  046134  013737  045232  052120          MOV     @#GECC2,@#DISK+<257.*2>;INSERT ECC2 ON DISK
 9642  046142  012605                          MOV     (SP)+,R5        ;;POP STACK INTO R5
 9643  046144  012604                          MOV     (SP)+,R4        ;;POP STACK INTO R4
 9644  046146  012603                          MOV     (SP)+,R3        ;;POP STACK INTO R3
 9645  046150  012602                          MOV     (SP)+,R2        ;;POP STACK INTO R2
 9646  046152  012601                          MOV     (SP)+,R1        ;;POP STACK INTO R1
 9647  046154  012600                          MOV     (SP)+,R0        ;;POP STACK INTO R0
 9648  046156  000207                          RTS     PC
 9649
 9650
 9651
 9652
 9653
 9654                                          .SBTTL  RH BASE ADDRESS CHANGE ROUTINE
 9655
 9656                                  ;**     THIS ROUTINE WILL ALLOW THE CHANGE OF THE BASE
 9657                                  ;**     ADDRESS FROM 176700 TO ANY TYPED VALUE
```

```
 9658
 9659  046160                          BASECH:
 9660  046160  104401  046166                  TYPE    ,65$            ;;TYPE ASCIZ STRING
 9661  046164  000425                          BR      64$             ;;GET OVER THE ASCIZ
 9662                                  ;;65$:  .ASCIZ  <15><12>/PRESENT BASE ADDRESS OF REGISTERS IS   /
 9663  046240                          64$:
 9664  046240  013746  001640                  MOV     @#RHCS1,-(SP)   ;GET READY TO TYPE OLD BASE
 9665  046244  104402                          TYPOC
 9666  046246  104401  046254                  TYPE    ,67$            ;;TYPE ASCIZ STRING
 9667  046252  000425                          BR      66$             ;;GET OVER THE ASCIZ
 9668                                  ;;67$:  .ASCIZ  <15><12>/TYPE NEW BASE ADDRESS FOLLOWED BY 'CR' /
 9669  046326                          66$:
 9670  046326  004737  054672                  JSR     PC,@#$TKINT     ;INITIALIZE THE TTY KEYBOARD
 9671  046332  104412                          RDOCT
 9672  046334  012700  001630                  MOV     #RHDB,R0        ;GET STARTING ADDRESS OF REGISTERS
 9673  046340  012701  000024                  MOV     #20.,R1         ;NUMBER OF REGISTERS
 9674  046344  042710  177700          1$:     BIC     #^C77,(R0)      ;CLEAR OLD BASE
 9675  046350  051620                          BIS     (SP),(R0)+      ;SET NEW BASE
 9676  046352  005301                          DEC     R1              ;COUNT
 9677  046354  001373                          BNE     1$              ;BRANCH IF 20 NOT DONE
 9678  046356  104401  046364                  TYPE    ,69$            ;;TYPE ASCIZ STRING
 9679  046362  000417                          BR      68$             ;;GET OVER THE ASCIZ
 9680                                  ;;69$:  .ASCIZ  <15><12>/PRESENT VECTOR ADDRESS IS /
 9681  046422                          68$:
 9682  046422  013746  001626                  MOV     @#RPVEC,-(SP)   ;GET READY TO TYPE OLD VECTOR ADDRESS
 9683  046426  104402                          TYPOC
 9684  046430  104401  046436                  TYPE    ,71$            ;;TYPE ASCIZ STRING
 9685  046434  000437                          BR      70$             ;;GET OVER THE ASCIZ
 9686                                  ;;71$:  .ASCIZ  <15><12>/TYPE NEW VECTOR ADDRESS OR RETYPE OLD ONE FOLLOWED BY "CR" /
 9687  046534                          70$:
 9688  046534  104412                          RDOCT
 9689  046536  012637  001626                  MOV     (SP)+,@#RPVEC   ;SETUP VECTOR ADDRESS
 9690  046542  104401  046550                  TYPE    ,73$            ;;TYPE ASCIZ STRING
 9691  046546  000402                          BR      72$             ;;GET OVER THE ASCIZ
 9692                                  ;;73$:  .ASCIZ  <15><12>/ /
 9693  046554                          72$:
 9694  046554  104401  046562                  TYPE    ,75$            ;;TYPE ASCIZ STRING
 9695  046560  000421                          BR      74$             ;;GET OVER THE ASCIZ
 9696                                  ;;75$:  .ASCIZ  <15><12>/RESTART PROGRAM FROM 200 OR 210/
 9697  046624                          74$:
 9698  046624  104401  046632                  TYPE    ,77$            ;;TYPE ASCIZ STRING
 9699  046630  000416                          BR      76$             ;;GET OVER THE ASCIZ
 9700                                  ;;77$:  .ASCIZ  <15><12>/NEW BASE WILL REMAIN   - /
 9701  046666                          76$:
 9702  046666  013746  001640                  MOV     @#RHCS1,-(SP)
 9703  046672  104402                          TYPOC
 9704  046674  104401  046702                  TYPE    ,79$            ;;TYPE ASCIZ STRING
 9705  046700  000416                          BR      78$             ;;GET OVER THE ASCIZ
 9706                                  ;;79$:  .ASCIZ  <15><12>/NEW VECTOR WILL REMAIN - /
 9707  046736                          78$:
 9708  046736  013746  001626                  MOV     @#RPVEC,-(SP)
 9709  046742  104402                          TYPOC
 9710  046744  104401  046752                  TYPE    ,81$            ;;TYPE ASCIZ STRING
 9711  046750  000416                          BR      80$             ;;GET OVER THE ASCIZ
 9712                                  ;;81$:  .ASCIZ  <15><12>/UNTIL PROGRAM IS RELOADED/
 9713  047006                          80$:
```

```
 9714  047006  000000                    HALT
 9715
 9716
 9717
 9718
 9719
 9720                          ;*THIS IS A LITTLE ROUTINE THAT TESTS NED BIT 11 IN RHCS2
 9721                          ;*THIS LOOPS HERE FOR EVER
 9722                          ;*TO BE USED ONLY IF DRIVES PRESENT LOOKING AT NED DOES NOT AGREE
 9723                          ;*WITH WHAT IS REALY THERE
 9724  047010  000000          ERUNIT: 0                          ;UNIT UNDER MANUAL TEST
 9725  047012  004737  042534  ERSTART:JSR     PC,@#CLDISK       ;SET GENERAL REG.
 9726  047016  013712  047010          MOV     @#ERUNIT,@R2      ;SELECT UNIT
 9727  047022  005714          1$:     TST     @R4               ;TEST RHER1
 9728  047024  032712  010000          BIT     #NED,@R2          ;TEST NED
 9729  047030  001401                  BEQ     2$                ;BRANCH IF GOOD
 9730  047032  000773                  BR      1$                ;NED NOT SET
 9731  047034  000772          2$:     BR      1$                ;NED SET
```

```
9732
9733                                    .SBTTL  DISK SIMULATION
9734                                ;;**********************************************************************
9735                                ;;**********************************************************************
9736                                ;*IN A WRITE HEADER AND DATA COMMAND FILL THE FOLLOWING
9737                                ;*WCLY=WITH CYLINDER TO BE ON DISK
9738                                ;*WSECTR=WITH SECTOR AND TRACK TO BE ON DISK
9739                                ;*WKEY1= WITH KEY1 TO BE ON DISK
9740                                ;*WKEY2= WITH KEY2 TO BE ON DISK
9741                                ;*FNWORD= NO OF DATA WORDS TO BE WRITTEN ON DISK
9742                                ;*THE COMMAND THEN IS   JSR PC,COMWHD
9743                                ;*
9744                                ;*
9745                                ;*
9746                                ;*
9747                                ;*IN A WRITE DATA COMMAND FILL THE FOLLOWING
9748                                ;*CYL=WITH CYLINDER TO BE FOUND ON DISK
9749                                ;*SECOTR= WITH SECTOR AND TRACK TO BE FOUND ON DISK
9750                                ;*KEY1= WITH KEY1 TO BE FOUND ON DISK
9751                                ;*KEY2= WITH KEY2 TO BE FOUND ON DISK
9752                                ;*X= 1  MUST BE ONE
9753                                ;*NOWORD= WITH NUMBER OF DATA WORDS TO BE WRITTEN
9754                                ;*THE COMMAND THEN IS JSR PC,COMHD
9755                                ;*
9756                                ;*
9757                                ;*
9758                                ;*
9759                                ;*
9760                                ;*
9761                                ;*IN A READ HEADER AND DATA COMMAND FILL THE FOLLOWING
9762                                ;*CYL= WITH CYLINDER TO BE FOUND ON DISK
9763                                ;*SECOTR= WITH SECTOR AND TRACK TO BE FOUND ON DISK
9764                                ;*KEY1= WITH KEY1 TO BE FOUND ON DISK
9765                                ;*KEY2=WITH KEY2 TO BE FOUND ON DISK
9766                                ;*DAWORD= WITH NUMBER OF WORDS TO BE FOUND ON DISK
9767                                ;*;X=0 MUST BE ZERO
9768                                ;*THE COMMAND THEN IS JSR PC,COMHD
9769                                ;*
9770                                ;*
9771                                ;*
9772                                ;*
9773                                ;*
9774                                ;*
9775                                ;*
9776                                ;*
9777                                ;*
9778                                ;*
9779                                ;*IN A READ DATA COMMAND FILL THE FOLLOWING
9780                                ;*CYL= WITH CYLINDER TO BE FOUND ON DISK
9781                                ;*SECOTR= WITH SECTOR AND TRACK TO BE FOUND ON DISK
9782                                ;*KEY1= WITH KEY1 TO BE FOUND ON DISK
9783                                ;*KEY2=WITH KEY2 TO BE FOUND ON DISK
9784                                ;*DAWORD= WITH NUMBER OF WORDS TO BE FOUND ON DISK
9785                                ;*;X=0 MUST BE ZERO
9786                                ;*THE COMMAND THEN IS JSR PC,COMHD
9787                                ;*
```

```
9788
9789
9790
9791
9792
9793
9794                    ;;*****************************************************************
9795                    ;*WRITE DATA COMMAND
9796                    ;*OR READ COMMAND I.E DATA ONLY OR HEADER AND DATA
9797                    ;;*****************************************************************
9798
9799
9800
9801
9802
9803                    ;**THIS SUBROUTINE IS THE FIRST IN A SERIES OF NESTED SUBROUTINES
9804
9805                    ;**IT ISSURE DIAGNOSTIC MODE, AN EXTRA DIAGNOSTIC INDEX, AND THE
9806                    ;**'GO' BIT
9807
9808                    ;**IT THEN CALL THREE OTHER SUBROUTINES, WHICH IN TURN CALL OTHER
9809                    ;**SUBROUTINES.  THESE ARE:
9810
9811                    ;**     SEARCH
9812                    ;**     RDHEAD
9813                    ;**     WRDATA
9814                    ;**     REDATA
9815
9816
9817
9818
9819
9820
9821  047036  000000            RUNCTR: .WORD   0
9822  047040  011637  002014    COMMD:  MOV     (SP),@#PCJSR    ;SAVE PC OF JSR + 4
9823  047044  162737  000004  002014   SUB     #4,@#PCJSR      ;SAVE PC OF JSR
9824  047052  010046                   MOV     R0,-(SP)        ;;PUSH R0 ON STACK
9825  047054  010146                   MOV     R1,-(SP)        ;;PUSH R1 ON STACK
9826  047056  010246                   MOV     R2,-(SP)        ;;PUSH R2 ON STACK
9827  047060  010346                   MOV     R3,-(SP)        ;;PUSH R3 ON STACK
9828  047062  010446                   MOV     R4,-(SP)        ;;PUSH R4 ON STACK
9829  047064  010546                   MOV     R5,-(SP)        ;;PUSH R5 ON STACK
9830
9831  047066  012777  000001  132564   MOV     #DMD,@RHMR      ;SET DIAGNOSTIC MODE
9832  047074  052777  000004  132556   BIS     #MINX,@RHMR     ;SET DIAGNOSTIC INDEX
9833  047102  042777  000004  132550   BIC     #MINX,@RHMR     ;CLEAR DIAGNOSTIC INDEX
9834  047110  052777  000001  132522   BIS     #GO,@RHCS1      ;ISSUE "GO" BIT & STALL 'TILL 'RUN'
9835                                                           ;(FUNCTION CODE IS ISSUED BY THE TEST)
9836  047116  012737  000113  047036  RUNWAT: MOV     #75.,@#RUNCTR   ;LOAD STALL COUNT = APPROX. 450US
9837                                                           ;FOR 11/50 CPU WITH CORE MEMORY
9838  047124  005337  047036    1$:     DEC     @#RUNCTR        ;COUNT DOWN ONE
9839  047130  001375                   BNE     1$              ;CONTINUE UNTIL = 0
9840
9841  047132  013746  047202           MOV     SECOTR,-(SP)    ;GET DESIRED SECTOR/TRACK
9842  047136  042716  177740           BIC     #177740,(SP)    ;MAKE ONLY SECTOR
9843  047142  012637  047152           MOV     (SP)+,  @#TRK   ;SAVE SECTOR
```

M2

```
9844  047146  004137  053246          JSR     R1,@#SEARCH     ;ISSUE SECTOR CLOCKS <---------------->
9845
9846  047152  000000          TPK:    .WORD   0
9847  047154  012701  000240          MOV     #^NOP,R1        ;GOING TO MOVE NOPS
9848  047160  010137  047212          MOV     R1,@#SSYN       ;NOP INTO SSYN
9849  047164  010137  047214          MOV     R1,@#HEDGAP     ;NOP INTO HEDGAP
9850  047170  010137  047216          MOV     R1,@#HEDSYN     ;NOP INTO HEDSYN
9851  047174  004137  047322          JSR     R1,@#RDHEAD
9852
9853
9854                                  ;*DUMMY ERROR CALL LOCATIONS FOR THE READ HEADER OPERATION
9855
9856  047200  000000          CYL:    .WORD   0               ;CYLINDER ADDRESS
9857  047202  000000          SECTTR: .WORD   0               ;SECTOR/TRACK ADDRESS
9858  047204  000000          KEY1:   .WORD   0               ;KEY1 WORD
9859  047206  000000          KEY2:   .WORD   0               ;KEY2 WORD
9860  047210  000000          X:      .WORD   0               ;X=1 WRITE COMMAND
9861                                                          ;X=0 READ COMMAND
9862
9863  047212  000240          SSYN:   NOP                     ;IF "ERROR 2" INSERTED BY RDHEAD
9864                                                          ;SUBROUTINE THEN THE FIRST SYNC.
9865                                                          ;IS NOT DETECTED.  NO BAD DATA
9866                                                          ;IS GIVEN BECAUSE SYNC=144000
9867                                                          ;CANNOT BE READ.  WORD NO
9868                                                          ;IS "1" BECAUSE THIS IS THE FIRST
9869                                                          ;WORD TESTED
9870
9871  047214  000240          HEDGAP: NOP                     ;IF "ERROR 3" INSERTED BY
9872                                                          ;RDHEAD SUBROUTINE THEN THE
9873                                                          ;HEADER GAP 0'S WERE NOT
9874                                                          ;WRITTEN RIGHT.
9875                                                          ;IF "WORD NO" CONTAINS, SAY
9876                                                          ;3(8), THEN IT IS THE THIRD
9877                                                          ;WORD OF A 5 WORD HEADER
9878                                                          ;GAP THAT IS WRONG
9879                                                          ;"BAD DATA" CONTAINS WHAT IS
9880                                                          ;GOING ON THE DISK
9881
9882  047216  000240          HEDSYN: NOP                     ;IF "ERROR 3" INSERTED BY RDHEAD
9883                                                          ;SUBROUTINE THEN THE HEADER SYNC.
9884                                                          ;GENERATED BY DCL IS WRONG
9885                                                          ;OR THE LAST BYTE
9886                                                          ;OF THE HEADER GAP 0'S IS WRONG
9887                                                          ;IN EITHER CASE WORD NO=6
9888                                                          ;RIGHT BYTE IS HEADER 0
9889                                                          ;LEFT BYTE IS SYNC
9890                                                          ;"BAD DATA" HAS WHAT IS GOING
9891                                                          ;ON DISK
9892
9893  047220  005737  002006          TST     @#ERFLGS        ;ARE ANY ERRORS DETECTED
9894  047224  001017                  BNE     OUT             ;IF YES, EXIT ---------->
9895  047226  005737  047210          TST     @#X             ;IS IT A DATA WRITE ?
9896  047232  001410                  BEQ     DAREAD          ;NO...THEN DO A DATA READ
9897  047234  005737  047314          TST     @#NOSYNC        ;IS THIS FORCED HEADER ERROR COMMAND
9898                                                          ;IF YES  NOSYNC=-1 THEN WRITE OR READ
9899                                                          ;IS SHUT OFF SO BRANCH OUT
```

```
 9900
 9901  047240  001011                        BNE     OUT             ;IF  NOSYNC=0 THEN CONTINUE
                                                                       ;EXIT IF SET ---------------------->
 9902
 9903  047242  004137  050566                JSR     R1,@#WRDATA     ;WRITE DATA <---------------------->
 9904
 9905  047246  000000            NOWORD: .WORD   0               ;NO OF WORDS TO BE WRITTEN
 9906  047250  000000            Y:      .WORD   0               ;
 9907  047252  000404                    BR      OUT             ;EXIT ---------------------------->
 9909
 9909  047254  004137  053522    DAREAD: JSR     R1,@#REDATA     ;READ DATA <---------------------->
 9910  047260  000000            DAWORD: .WORD   0               ;NO OF WORDS TO BE READ
 9911  047262  000000                    .WORD   0
 9912  047264                    OUT:
 9913  047264  012605                    MOV     (SP)+,R5        ;;POP STACK INTO R5
 9914  047266  012604                    MOV     (SP)+,R4        ;;POP STACK INTO R4
 9915  047270  012603                    MOV     (SP)+,R3        ;;POP STACK INTO R3
 9916  047272  012602                    MOV     (SP)+,R2        ;;POP STACK INTO R2
 9917  047274  012601                    MOV     (SP)+,R1        ;;POP STACK INTO R1
 9918  047276  012600                    MOV     (SP)+,R0        ;;POP STACK INTO R0
 9919  047300  000207                    RTS     PC              ;EXIT ROUTINE
```

```
 9920
 9921                                   ;;*****************************************************************
 9922                                   ;*THE DISK SECTOR IS DEVIDED AS FOLLOWS
 9923                                   ;*19 WORDS OF 0, ONE WORD 144000
 9924                                   ;*THESE MAKE 39 BYTES FOR SECTOR GAP AND ONE SYNC. BYTE
 9925
 9926
 9927  047302  014400                   RSYNC:  14400
 9928  047304  000000                   RCYL:   0
 9929  047306  000000                   RSETR:  0
 9930  047310  000000                   RKEY1:  0
 9931  047312  000000                   RKEY2:  0
 9932
 9933
 9934                                   ;*5 WORDS OF 0 ONE WORD 144000
 9935                                   ;*THESE MAKE 11 BYTES FOR HEADER GAP AND ONE SYNC. BYTE
 9936                                   ;*THESE ARE DCL GENERATED
 9937
 9938
 9939
 9940                                   ;*THERE ARE 256 WORDS OF DATA
 9941                                   ;*THERE ARE 2 WORDS FOR ECC GENERATED BY DCL
 9942                                   ;*15 WORDS OF 0 FOR DATA GAP AND TOLERANCE GAP
 9943                                   ;;*****************************************************************
 9944
 9945
```

```
 9946
 9947
 9948
 9949
 9950                                    ;;*****************************************************************
 9951                                    ;*READ DISK HEADER
 9952                                    ;;*****************************************************************
 9953
 9954
 9955
 9956
 9957
 9958  047314  000000                    NOSYNC: 0                       ;FORCED HEADER ERROR = -1
 9959                                                                    ;NORMAL = 0
 9960  047316  000000                    TY:     0                       ;ERROR TYPE NO.
 9961  047320  000000                    ERWORD: 0                       ;ERROR WORD NO.
 9962
 9963
 9964
 9965
 9966  047322  012137  047304            RDHEAD: MOV     (R1)+,  @#RCYL  ;STORE CYLINDER ADDRESS
 9967  047326  012137  047306                    MOV     (R1)+,  @#RSETR ;STORE SECTOR AND TRACK ADDRESS
 9968  047332  012137  047310                    MOV     (R1)+,  @#RKEY1 ;STORE KEY1
 9969  047336  012137  047312                    MOV     (R1)+,  @#RKEY2 ;STORE KEY2
 9970  047342  012137  050112                    MOV     (R1)+,  @#COMPA ;STORE COMPARE OR NOT
 9971  047346  010146                            MOV     R1,-(SP)        ;;PUSH R1 ON STACK
 9972  047350  013700  001660                    MOV     @#RHMR, R0      ;R0 CONTAINS MAINTANENCE REG.
 9973  047354  012705  000002                    MOV     #2,     R5      ;R5 IS A COUNTER FOR WORDS
 9974  047360  012710  000001                    MOV     #DMD,   @R0     ;DIAG. MODE
 9975  047364  052710  000010                    BIS     #MSTCK,@R0      ;SET SECTOR FOR FIRST WORD
 9976  047370  052710  000002                    BIS     #MCLK,@R0       ;SET CLOCK FOR FIRST WORD
 9977  047374  042710  000012                    BIC     #MSTCK!MCLK,@R0 ;RESET SECTOR AND CLOCK
 9978  047400  000404                            BR      2$              ;BRANCH OVER GIVING SECTOR FOR FIRST TIME
 9979  047402  012710  000013            1$:     MOV     #MSTCK!MCLK!DMD,@R0;SET SECTOR, CLOCK, DIAG. MODE, RESET INDEX
 9980  047406  042710  000012                    BIC     #MSTCK!MCLK,@R0 ;RESET SECTOR, CLOCK
 9981  047412  012702  000007            2$:     MOV     #7,     R2      ;R2 IS A COUNTER FOR BYTES
 9982  047416  052710  000002            3$:     BIS     #MCLK,  @R0     ;SET CLOCK
 9983  047422  042710  000002                    BIC     #MCLK,  @R0     ;RESET CLOCK
 9984  047426  005302                            DEC     R2              ;BYTE COUNTER
 9985  047430  001372                            BNE     3$              ;BRANCH IF BYTE NOT COMPLETE
 9986  047432  005305                            DEC     R5              ;WORD COUNTER
 9987  047434  001362                            BNE     1$              ;BRANCH IF WORD NOT COMPLETE
 9988  047436  012702  000022                    MOV     #18.,   R2      ;NO OF WORDS OF ZEROS
 9989  047442  005037  050110            4$:     CLR     @#WORD          ;READ 0
 9990  047446  004737  050114                    JSR     PC,     @#READ  ;GO TO READ
 9991  047452  005302                            DEC     R2              ;COUNT
 9992  047454  001372                            BNE     4$
 9993  047456  013737  047302  050110            MOV     @#RSYNC,@#WORD  ;SYNC. WORD
 9994  047464  004737  050114                    JSR     PC,     @#READ
 9995  047470  032710  001000                    BIT     #DTSY,  @R0     ;SYNC. BYTE DETECTED?
 9996  047474  001012                            BNE     5$              ;BRANCH IF SYNC DETECTED
 9997  047476  012737  000001  047320            MOV     #1,     @#ERWORD        ;ERROR WORD NO
 9998  047504  013737  047302  001124            MOV     @#RSYNC,@#$GDDAT        ;SYNC WORD
 9999  047512  012737  104002  047212            MOV     #104002,@#SSYN  ;INSERT "ERROR 2" IN SSYN
10000  047520  000571                            BR      13$             ;BRANCH OUT
10001  047522  013737  047304  050110  5$:     MOV     @#RCYL, @#WORD  ;SETUP CYLINDER
```

```
10002  047530  004737  050114                      JSR     PC,     @#READ  ;READ
10003  047534  013737  047306  050110              MOV     @#RSETR,@#WORD  ;SETUP SECTOR/TRACK
10004  047542  004737  050114                      JSR     PC,@#READ       ;READ
10005  047546  013737  047310  050110              MOV     @#RKEY1,@#WORD  ;SETUP KEY1
10006  047554  004737  050114                      JSR     PC,@#READ       ;READ
10007  047560  013737  047312  050110              MOV     @#RKEY2,@#WORD  ;SETUP KEY2
10008  047566  004737  050114                      JSR     PC,@#READ       ;READ
10009  047572  013737  051100  050110              MOV     @#WCRC,@#WORD   ;SETUP CRC
10010  047600  004737  050114                      JSR     PC,@#READ       ;READ
10011  047604  005737  002026                      TST     @#TESDTE        ;IS THIS A DRIVE TIMING ERROR
10012  047610  001135                              BNE     13$             ;BRANCH OUT IF YES
10013  047612  005737  050112                      TST     @#COMPA         ;IS THIS A READ OR WRITE COMMAND
10014  047616  001472                              BEQ     11$
10015  047620  012705  051102                      MOV     #HEGAP, R5      ;POINTER FOR HEADER GAP
10016  047624  012702  000005                      MOV     #5,     R2      ;NO OF WORDS OF ZEROS
10017  047630  012737  000006  047320  6$:         MOV     #6,@#ERWORD     ;ERROR WORD NO SET
10018  047636  004737  050346                      JSR     PC,@#WRITE      ;FOR HEADER GAP
10019  047642  005737  050344                      TST     @#WWORD         ;TEST WRITTEN WORD
10020  047646  001413                              BEQ     7$              ;BRANCH IF GOOD THAT IS 0
10021  047650  160237  047320                      SUB     R2,@#ERWORD     ;WORD NO IN ERROR
10022  047654  005037  001124                      CLR     @#$GDDAT                ;GOOD WORD SHOULD BE 0
10023  047660  013737  050344  001126              MOV     @#WWORD,        $BDDAT  ;BAD DATA
10024  047666  012737  104003  047214              MOV     #104003,@#HEDGAP        ;"ERROR 2" GOES IN HEDGAP
10025  047674  000503                              BR      13$             ;BRANCH OUT
10026  047676  013725  050344          7$:         MOV     @#WWORD,(R5)+   ;SAVE HEADER GAP
10027  047702  005302                              DEC     R2
10028  047704  001351                              BNE     6$
10029  047706  004737  050346                      JSR     PC,     @#WRITE ;WRITE HEADER (DATA) GAP SYNC
10030  047712  023737  047302  050344              CMP     @#RSYNC,@#WWORD
10031  047720  001426                              BEQ     10$
10032  047722  005737  047314                      TST     @#NOSYNC        ;IS THIS FORCED HEADER ERROR COMMAND
10033                                                                      ;IF YES  NOSYNC=-1 THEN WRITE OR READ
10034                                                                      ;IS SHUT OFF SO BRANCH OUT
10035                                                                      ;IF NO NOSYNC=0 THEN CONTINUE
10036  047726  001406                              BEQ     14$             ;BRANCH IF TRUE ERROR
10037  047730  005737  050344                      TST     @#WWORD         ;
10038  047734  001420                              BEQ     10$             ;BRANCH IF GOOD
10039  047736  005037  001124                      CLR     @#$GDDAT        ;IT SHOULD BE ZERO
10040  047742  000403                              BR      15$             ;BRANCH TO TYPE ERROR
10041  047744  013737  047302  001124  14$:         MOV     @#RSYNC,@#$GDDAT        ;GOOD DATA
10042  047752  013737  050344  001126  15$:         MOV     @#WWORD,@#$BDDAT        ;BAD DATA
10043  047760  012737  000006  047320              MOV     #6,     @#ERWORD
10044  047766  012737  104003  047216              MOV     #104003,@#HEDSYN
10045  047774  000443                              BR      13$             ;BRANCH OUT
10046  047776  013725  050344          10$:        MOV     @#WWORD,(R5)+   ;SAVE DATA SYNC.
10047  050002  000440                              BR      13$
10048                                  ;*READ COMMAND START FROM HERE
10049  050004  012702  000005          11$:        MOV     #5,     R2
10050  050010  005037  050110          12$:        CLR     WORD
10051  050014  004737  050114                      JSR     PC,     READ    ;READ HEADER GAP
10052  050020  005302                              DEC     R2              ;IS 5 HEADER GAP ZEROS COMPLETE
10053  050022  001372                              BNE     12$             ;IF NOT BRANCH
10054  050024  013737  047302  050110              MOV     @#RSYNC,@#WORD  ;SYNC WORD
10055  050032  004737  050114                      JSR     PC,     READ    ;READ HEADER (DATA) SYNC)
10056  050036  005737  047314                      TST     @#NOSYNC
10057  050042  001404                              BEQ     16$             ;IF NOT ERROR COMMAND BRANCH
```

```
10058  050044  032710  001000              BIT     #DTSY,@R0       ;SYNC. DETECTED
10059  050050  001415                      BEQ     13$             ;IF ZERO BRANCH OUT
10060  050052  000403                      BR      17$             ;IF NOT ZERO BRANCH TO ERROR
10061  050054  032710  001000        16$:  BIT     #DTSY,  @R0     ;SYNC. DETECTED?
10062  050060  001011                      BNE     13$             ;BRANCH IF YES
10063  050062  012737  000006  047320 17$:  MOV     #6,@#ERWORD     ;ERROR WORD NO.
10064  050070  013737  047302  001124       MOV     @#RSYNC,@#$GDDAT;SYNC WORD
10065  050076  012737  104002  047216       MOV     #104002,@#HEDSYN
10066  050104                      13$:
10067  050104  012601                      MOV     (SP)+,R1        ;;POP STACK INTO R1
10068  050106  000201                      RTS     R1
10069
10070
```

```
10071
10072
10073
10074
10075
10076
10077                                   ;;*****************************************************************
10078                                   ;*READ ONE WORD IN "WORD"
10079                                   ;;*****************************************************************
10080
10081
10082
10083
10084  050110  000000                   WORD:   0
10085  050112  000000                   COMPA:  0
10086
10087
10088
10089
10090  050114                           READ:
10091  050114  010246                           MOV     R2,-(SP)        ;;PUSH R2 ON STACK
10092  050116  012705  000002                   MOV     #2,     R5      ;WORD COUNTER
10093  050122  012710  000001                   MOV     #DMD,   @R0     ;SET DIAG. MODE
10094  050126  006037  050110                   ROR     @#WORD          ;CHECKING IF THERE IS A ONE
10095  050132  103002                           BCC     1$              ;IF NO ONE BRANCH
10096  050134  052710  000020                   BIS     #MRD,   @R0     ;SET BIT 4 IF DATA HAS ONE
10097  050140  012702  000007           1$:     MOV     #7,     R2      ;BYTE COUNTER
10098  050144  052710  000012                   BIS     #MSTCK!MCLK,    @R0     ;SET CLOCK,DATA IF ANY, SECTOR
10099  050150  005737  045234                   TST     @#TSECCG        ;IS THIS BIT TO GENERATE AND TEST ECC
10100  050154  001411                           BEQ     6$              ;BRANCH IF NO
10101  050156  032710  000020                   BIT     #MRD,@R0        ;IS DATA BIT A ONE
10102  050162  001404                           BEQ     5$              ;BRANCH IF DATA BIT IS 0
10103  050164  012737  177777  045226           MOV     #-1,@#ECDATA    ;ECC DATA BIT IS A ONE
10104  050172  000402                           BR      6$              ;BRANCH
10105  050174  005037  045226           5$:     CLR     @#ECDATA        ;ECC DATA BIT IS A 0
10106  050200  012746  000001           6$:     MOV     #DMD,   -(SP)   ;KEEP ONLY DIAG. MODE
10107  050204  006037  050110                   ROR     @#WORD          ;CHECKING IF THERE IS A ONE
10108  050210  103002                           BCC     2$              ;IF NO ONE BRANCH
10109  050212  012716  000021                   MOV     #MRD!DMD,       (SP)    ;KEEP DATA AND DIAG. MODE
10110  050216  012610                   2$:     MOV     (SP)+,  @R0     ;PUT IN DATA,RESET CLOCK, SECTOR
10111  050220  005737  045234                   TST     @#TSECCG        ;IS ECC TO BE GENERATED FOR THIS BIT
10112  050224  001404                           BEQ     3$              ;BRANCH IF NO
10113  050226  005237  045246                   INC     @#DATENV        ;NUMBER OF CLOCKS GIVEN FOR DATA ENVELOPE
10114  050232  004737  045264                   JSR     PC,@#ECTEST     ;GO TO GENERATE AND TEST ECC
10115  050236  052710  000002           3$:     BIS     #MCLK,  @R0     ;SET CLOCK
10116  050242  005737  045234                   TST     @#TSECCG        ;IS THIS BIT TO GENERATE ECC
10117  050246  001411                           BEQ     8$              ;BRANCH IF NO
10118  050250  032710  000020                   BIT     #MRD,@R0        ;IS DATA BIT A ONE
10119  050254  001404                           BEQ     7$              ;BRANCH IF DATA BIT IS = 0
10120  050256  012737  177777  045226           MOV     #-1,@#ECDATA    ;ECC DATA BIT IS A ONE
10121  050264  000402                           BR      8$              ;BRANCH
10122  050266  005037  045226           7$:     CLR     @#ECDATA        ;ECC DATA BIT IS = 0
10123  050272  012746  000001           8$:     MOV     #DMD,   -(SP)   ;KEEP DIAG. MODE
10124  050276  006037  050110                   ROR     @#WORD          ;CHECKING IF THERE IS A ONE
10125  050302  103002                           BCC     4$              ;BRANCH IF NO ONE
10126  050304  012716  000021                   MOV     #MRD!DMD,(SP)   ;KEEP DIAG. MODE AND DATA
```

```
10127  050310  012610                  4$:     MOV     (SP)+,  @R0     ;SET DATA, DIAG. MODE, CLEAR CLOCK
10128  050312  005737  045234                  TST     @#TSECCG        ;IS THIS BIT TO GENERATE ECC
10129  050316  001404                          BEQ     9$              ;BRANCH IF NO
10130  050320  005237  045246                  INC     @#DATENV        ;NUMBER OF CLOCKS GIVEN FOR DATA ENVELOPE
10131  050324  004737  045264                  JSR     PC,@#ECTEST     ;GO TO GENERATE AND TEST ECC
10132  050330  005302                  9$:     DEC     R2              ;BYTE COUNTER
10133  050332  001341                          BNE     3$              ;BRANCH IF ONE BYTE NOT COMPLETE
10134  050334  005305                          DEC     R5              ;WORD COUNTER
10135  050336  001300                          BNE     1$              ;BRANCH IF ONE WORD NOT COMPLETE
10136  050340  012602                          MOV     (SP)+,R2        ;;POP STACK INTO R2
10137  050342  000207                          RTS     PC
10138
10139
```

```
10140
10141
10142
10143
10144                                   ;;*****************************************************************
10145                                   ;*WRITE ONE WORD WHICH COMES BACK IN "WWORD"
10146                                   ;;*****************************************************************
10147
10148
10149
10150
10151
10152
10153  050344  000000                   WWORD:  0
10154
10155
10156
10157
10158  050346                           WRITE:
10159  050346  010046                           MOV     R0,-(SP)        ;;PUSH R0 ON STACK
10160  050350  010246                           MOV     R2,-(SP)        ;;PUSH R2 ON STACK
10161  050352  010346                           MOV     R3,-(SP)        ;;PUSH R3 ON STACK
10162  050354  010546                           MOV     R5,-(SP)        ;;PUSH R5 ON STACK
10163  050356  012705  000002                   MOV     #2,R5           ;WORD COUNTER
10164  050362  012710  000001                   MOV     #1,@R0          ;SET DIAG. MODE
10165                                                                    ;R0 HAS RHMR ADDRESS IN IT
10166  050366  012702  000007           1$:     MOV     #7,R2           ;BYTE COUNTER
10167  050372  012710  000013                   MOV     #MSTCK!MCLK!DMD,@R0;SET SECTOR AND CLOCK
10168  050376  032710  000040                   BIT     #MWR,@R0        ;CHECK WRITE BIT IN MAINT. REG.
10169  050402  001406                           BEQ     2$              ;BRANCH IF ZERO
10170  050404  012737  177777  045226           MOV     #-1,@#ECDATA    ;ECC DATA BIT IS A ONE
10171  050412  000261                           SEC                     ;SET CARRY
10172  050414  006003                           ROR     R3              ;MOVE 1 FORWARD
10173  050416  000404                           BR      3$
10174  050420  005037  045226           2$:     CLR     @#ECDATA        ;ECC DATA BIT IS = 0
10175  050424  000241                           CLC                     ;CLEAR CARRY
10176  050426  006003                           ROR     R3              ;MOVE 0 FOR WWORD
10177  050430  012710  000001           3$:     MOV     #DMD,@R0        ;CLEAR SECTOR AND CLOCK
10178  050434  005737  045234                   TST     @#TSECCG        ;IS THIS BIT TO GENERATE ECC
10179  050440  001404                           BEQ     4$              ;BRANCH IF NO
10180  050442  005237  045246                   INC     @#DATENV        ;NUMBER OF CLOCKS GIVEN FOR DATA ENVELOPE
10181  050446  004737  045264                   JSR     PC,@#ECTEST     ;GO TO GENERATE AND TEST ECC
10182  050452  052710  000002           4$:     BIS     #MCLK,@R0       ;SET CLOCK IN RHMR
10183  050456  032710  000040                   BIT     #MWR,@R0        ;CHECK WRITE BIT IN RHMR
10184  050462  001406                           BEQ     5$              ;BRANCH IF ZERO
10185  050464  012737  177777  045226           MOV     #-1,@#ECDATA    ;ECC DATA BIT IS A ONE
10186  050472  000261                           SEC                     ;SET CARRY
10187  050474  006003                           ROR     R3              ;MOVE 1 FOR WWORD
10188  050476  000404                           BR      6$
10189  050500  005037  045226           5$:     CLR     @#ECDATA        ;ECC DATA BIT IS ZERO
10190  050504  000241                           CLC                     ;CLEAR CARRY
10191  050506  006003                           ROR     R3              ;MOVE 0 FOR WWORD
10192  050510  012710  000001           6$:     MOV     #DMD,   @R0     ;CLEAR CLOCK
10193  050514  005737  045234                   TST     @#TSECCG        ;IS THIS BIT TO GENERATE ECC
10194  050520  001404                           BEQ     7$              ;BRANCH IF NO
10195  050522  005237  045246                   INC     @#DATENV        ;NUMBER OF CLOCKS GIVEN FOR DATA ENVELOPE
```

```
10196  050526  004737  045264                JSR     PC,@#ECTEST     ;GO TO GENERATE AND TEST ECC
10197  050532  005302                  7$:   DEC     R2              ;COUNT FOR BYTE END
10198  050534  001346                        BNE     4$              ;IF NOT BYTE END BRANCH
10199  050536  005305                        DEC     R5              ;COUNT FOR WORD END
10200  050540  001312                        BNE     1$              ;IF NOT WORD END BRANCH
10201
10202  050542  010337  050344                MOV     R3,@#WWORD      ;STORE THE WORD
10203
10204  050546  012605                        MOV     (SP)+,R5        ;;POP STACK INTO R5
10205  050550  012603                        MOV     (SP)+,R3        ;;POP STACK INTO R3
10206  050552  012602                        MOV     (SP)+,R2        ;;POP STACK INTO R2
10207  050554  012600                        MOV     (SP)+,R0        ;;POP STACK INTO R0
10208  050556  000207                        RTS     PC
10209
10210
```

```
10211
10212
10213
10214
10215                                   ;;*****************************************************************
10216                                   ;*WRITE DATA - PUT DATA INTO "DISK" AREA FROM "WWORD"
10217                                   ;*ONE WORD AT A TIME
10218                                   ;;*****************************************************************
10219
10220
10221
10222
10223
10224
10225
10226  050560  000000                   COUNTD: 0
10227  050562  000400                   FORMAT: 256.
10228  050564  000000                   ZWORDS: 0
10229  050566                           WRDATA:
10230  050566  011137  050560                   MOV     (R1),@#COUNTD   ;STORE NO. OF WORDS TO BE WRITTEN
10231  050572  012102                           MOV     (R1)+,R2        ;SAME IN R2
10232  050574  012137  050112                   MOV     (R1)+,@#COMPA   ;COMPARE OR NOT
10233
10234  050600  010046                           MOV     R0,-(SP)        ;;PUSH R0 ON STACK
10235  050602  010146                           MOV     R1,-(SP)        ;;PUSH R1 ON STACK
10236  050604  010246                           MOV     R2,-(SP)        ;;PUSH R2 ON STACK
10237  050606  010346                           MOV     R3,-(SP)        ;;PUSH R3 ON STACK
10238  050610  010446                           MOV     R4,-(SP)        ;;PUSH R4 ON STACK
10239
10240  050612  012701  000016                   MOV     #14.,R1         ;NO. OF TOLERANCE GAP WORDS
10241  050616  012703  052124                   MOV     #TOLGAP,R3      ;START OF TOLERANCE GAP TABLE
10242  050622  012723  177777           1$:     MOV     #-1,(R3)+       ;MAKE IT 177777
10243  050626  005301                           DEC     R1              ;IS 14 COMPLETED
10244  050630  001374                           BNE     1$              ;IF NO BRANCH
10245  050632  013700  001660                   MOV     @#RHMR, R0      ;R0 CONTAINS MAINTANENCE REG.
10246  050636  013746  050562                   MOV     @#FORMAT,-(SP)
10247  050642  163716  050560                   SUB     @#COUNTD,(SP)
10248  050646  011637  050564                   MOV     (SP),@#ZWORDS   ;NO. OF ZERO WORDS TO BE WRITTEN
10249  050652  012604                           MOV     (SP)+,R4
10250  050654  005737  002024                   TST     @#TSECC         ;IS THIS AN ECC TEST ?
10251  050660  001403                           BEQ     7$              ;BRANCH IF NO
10252  050662  012737  177777  045234           MOV     #-1,@#TSECCG    ;THESE BITS ARE TO GENERATE ECC
10253  050670  012703  051116           7$:     MOV     #DISK,R3        ;ADDRESS THE "DISK" AREA
10254  050674  004737  050346           2$:     JSR     PC,@#WRITE      ;WRITE INTO "WWORD"
10255  050700  013723  050344                   MOV     @#WWORD,(R3)+   ;STORE ON SIMULATED DISK
10256  050704  005302                           DEC     R2              ;COUNT DOWN
10257  050706  001372                           BNE     2$              ;CONTINUE IF ALL WORDS NOT WRITTEN
10258  050710  005704                           TST     R4              ;ANY ZEROS TO BE WRITTEN ?
10259  050712  001406                           BEQ     4$              ;BRANCH IF NONE TO BE WRITTEN
10260  050714  004737  050346           3$:     JSR     PC,@#WRITE      ;WRITE ZEROS INTO "WWORD"
10261  050720  013723  050344                   MOV     @#WWORD,(R3)+   ;STORE INTO "DISK"
10262  050724  005304                           DEC     R4
10263  050726  001372                           BNE     3$
10264  050730  005037  045234           4$:     CLR     @#TSECCG        ;NO MORE ECC TO BE GENERATED
10265  050734  012701  000002                   MOV     #2,R1
10266  050740  004737  050346           5$:     JSR     PC,WRITE        ;WRITE ECC1 AND ECC2 ON SIMULATED DISK
```

K3

```
10267  050744  013723  050344                  MOV     @#WWORD,(R3)+   ;STORE ON WEEC1 AND WEEC2
10268  050750  005301                          DEC     R1
10269  050752  001372                          BNE     5$
10270  050754  004737  050346                  JSR     PC,WRITE        ;WRITE DATA GAP INTO "WWORD"
10271  050760  013723  050344                  MOV     @#WWORD,(R3)+   ;STORE INTO "DISK"
10272  050764  012701  000016                  MOV     #14.,R1
10273  050770  004737  050346          6$:     JSR     PC,@#WRITE      ;WRITE TOLERANCE GAP ZEROS
10274  050774  013723  050344                  MOV     @#WWORD,(R3)+   ;STORE INTO "DISK"
10275  051000  005301                          DEC     R1
10276  051002  001372                          BNE     6$
10277
10278  051004  012604                          MOV     (SP)+,R4        ;;POP STACK INTO R4
10279  051006  012603                          MOV     (SP)+,R3        ;;POP STACK INTO R3
10280  051010  012602                          MOV     (SP)+,R2        ;;POP STACK INTO R2
10281  051012  012601                          MOV     (SP)+,R1        ;;POP STACK INTO R1
10282  051014  012600                          MOV     (SP)+,R0        ;;POP STACK INTO R0
10283
10284  051016  000201                          RTS     R1
10285
10286
```

```
10287
10288
10289
10290
10291
10292
10293                           ;;*****************************************************************
10294                           ;*WRITE HEADER AND DATA
10295 -                         ;*
10296                           ;*
10297                           ;*THIS IS THE SIMULATED DISK
10298                           ;*ONLY ONE SECTOR OF SPACE IS ALLOWED
10299                           ;;*****************************************************************
10300
10301
10302
10303
10304
10305
10306
10307  051020  000023           SECGAP: .BLKW   19.              ;SECTOR GAP 38 BYTES OF 0
10308  051066  000001           WSSYNC: .BLKW   1               ;SECTOR GAP 1 BYTE OF 0 ONE SYNC BYTE
10309  051070  000004           HEADER: .BLKW   4               ;HEADER = CYL, SECTOR/TRACK, KEY1, KEY2
10310  051100  000001           WCRC:   .BLKW   1               ;CRC
10311  051102  000005           HEGAP:  .BLKW   5               ;HEADER GAP 10 BYTES OF 0
10312  051114  000001           HDWSYN: .BLKW   1               ;HEADER GAP 1 BYTE OF 0 ONE SYNC. BYTE
10313  051116                   SILOTB:                 ;USED IN SILO TEST AS SILO TABLE
10314  051116  000400           DISK:   .BLKW   256.            ;DATA SPACE
10315  052116  000001           WECC1:  .BLKW   1               ;ECC1
10316  052120  000001           WECC2:  .BLKW   1               ;ECC2
10317  052122  000001           DTAGAP: .BLKW   1               ;DATA GAP 2 BYTES OF 0
10318  052124  000016           TOLGAP: .BLKW   14.             ;TOLERANCE GAP 28 BYTES OF 0
10319
10320
```

M3

```
10321
10322
10323
10324
10325                           ;;*****************************************************************
10326                           ;*WRITE HEADER AND DATA
10327                           ;;*****************************************************************
10328
10329
10330
10331
10332                           ;**THIS SUBROUTINE IS THE FIRST IN A SERIES OF NESTED SUBROUTINES
10333
10334                           ;**IT ISSUES DIAGNOSTIC MODE, AN EXTRA DIAGNOSTIC INDEX, AND THE
10335                           ;***'GO' BIT
10336
10337                           ;**IT THEN CALL THREE OTHER SUBROUTINES, WHICH IN TURN CALL OTHER
10338                           ;**SUBROUTINES.  THESE ARE:
10339
10340                           ;**     SEARCH
10341                           ;**     WRHEAD
10342                           ;**     WRDATA
10343
10344
10345
10346
10347
10348
10349
10350  052160  000000           RNCTR1: .WORD   0               ;'RUN' LINE STALL COUNTER
10351  052162  011637  002014   COMWHD: MOV     (SP),@#PCJSR    ;SAVE PC OF JSR + 4
10352  052166  162737  000004  002014  SUB     #4,@#PCJSR      ;SAVE PC OF JSR
10353  052174  010046                  MOV     R0,-(SP)        ;;PUSH R0 ON STACK
10354  052176  010146                  MOV     R1,-(SP)        ;;PUSH R1 ON STACK
10355  052200  010246                  MOV     R2,-(SP)        ;;PUSH R2 ON STACK
10356  052202  010346                  MOV     R3,-(SP)        ;;PUSH R3 ON STACK
10357  052204  010446                  MOV     R4,-(SP)        ;;PUSH R4 ON STACK
10358  052206  010546                  MOV     R5,-(SP)        ;;PUSH R5 ON STACK
10359
10360  052210  012777  000001  127442  MOV     #DMD,@RHMR      ;SET DIAGNOSTIC MODE
10361  052216  052777  000004  127434  BIS     #MINX,@RHMR     ;SET DIAGNOSTIC INDEX
10362  052224  042777  000004  127426  BIC     #MINX,@RHMR     ;CLEAR IT
10363  052232  052777  000001  127400  BIS     #GO,@RHCS1      ;SET 'GO' BIT & STALL 'TILL 'RUN'
10364
10365  052240  012737  000113  052160  RNWAT1: MOV     #75.,@#RNCTR1   ;LOAD STALL COUNTER = APPROX. 450US
10366                                                          ;FOR 11/50 CPU WITH CORE MEMORY
10367  052246  005337  052160          1$:     DEC     @#RNCTR1        ;COUNT DOWN 1 TIME
10368  052252  001375                  BNE     1$              ;CONTINUE UNTIL 0
10369
10370  052254  013746  052340          MOV     @#WSECTR,-(SP)  ;GET DESIRED SECTOR/TRACK
10371  052260  042716  177740          BIC     #177740,(SP)    ;MAKE ONLY SECTOR
10372  052264  012637  052274          MOV     (SP)+,@#WTRK    ;SAVE SECTOR
10373
10374  052270  004137  053246          JSR     R1,@#SEARCH     ;ISSUE SECTOR CLOCKS <------------>
10375  052274  000000          WTRK:   .WORD   0               ;SECTOR NO.
10376
```

```
10377  052276  012701  000240                MOV     #+NOP,R1        ;GOING TO MOVE NOPS
10378  052302  010137  052350                MOV     R1,@#SEGPER     ;NOP INTO SEGAP
10379  052306  010137  052352                MOV     R1,@#FSYNER     ;NOP INTP FSYNER
10380  052312  010137  052354                MOV     R1,@#ERHEAD     ;NOP INTO ERHEAD
10381  052316  010137  052356                MOV     R1,@#ERCRC      ;NOP INTO ERCRC
10382  052322  010137  052360                MOV     R1,@#ERHDGP     ;NOP INTO ERHDGAP
10383  052326  010137  052362                MOV     R1,@#HDESYN     ;NOP INTO HDESYN
10384
10385  052332  004137  052432                JSR     R1,@#WRHEAD     ;WRITE THE HEADER <---------------->
10386  052336  000000            WCYL:   0               ;CYLINDER
10387  052340  000000            WSECTR: 0               ;SECTOR AND TRACK
10388  052342  000000            WKEY1:  0               ;KEY1
10389  052344  000000            WKEY2:  0               ;KEY2
10390  052346  000000            GCRC:   0               ;GOOD CRC
10391
10392                                    ;*DUMMY ERROR CALL LOCATIONS FOR THE WRITE HEADER OPERATION
10393
10394
10395  052350  000240            SEGPER: NOP             ;IF "ERROR 6" INSERTED BY
10396                                                    ;WRHEAD SUBROUTINE THEN
10397                                                    ;SECTOR GAP GOING ON DISK
10398                                                    ;IS NOT RIGHT.
10399
10400                                                    ;WORD NO. CONTAINS WHICH
10401                                                    ;WORD IS WRONG, THAT IS
10402                                                    ;FIRST OF TENTH OR WHAT EVER NO.
10403                                                    ;BAD WORD IS GOING ON DISK
10404
10405  052352  000240            FSYNER: NOP             ;IF "ERROR 6" INSERTED BY
10406                                                    ;WRHEAD SUBROUTINE THEN
10407                                                    ;THE LAST 0 BYTE OF SECTOR
10408                                                    ;GAP, OR FIRST SYNC. BYTE
10409                                                    ;AFTER SECTOR GAP IS IN
10410                                                    ;ERROR.
10411
10412                                                    ;WORD NO. CONTAINS 20
10413                                                    ;RIGHT BYTE IS SECTOR GAP
10414                                                    ;LEFT BYTE IS SYNC. BYTE
10415                                                    ;BAD WORD IS WHAT IS GOING ON
10416                                                    ;DISK.
10417
10418  052354  000240            ERHEAD: NOP             ;IF "ERROR 6" INSERTED BY
10419                                                    ;WRHEAD SUBROUTINE THEN
10420                                                    ;HEADER GOING ON DISK
10421                                                    ;IS WRONG.
10422
10423                                                    ;WORD NO 1 = CYLINDER NO
10424                                                    ;WORD NO 2 = SECTOR/TRACK
10425                                                    ;WORD NO 3 = KEY1
10426                                                    ;WORD NO 4 = KEY2
10427                                                    ;BAD WORD IS WHAT IS GOING ON
10428                                                    ;DISK
10429
10430  052356  000240            ERCRC:  NOP
10431                                                    ;IF "ERROR 6" INSERTED BY
10432                                                    ;WRHEAD SUBROUTINE THEN CRC WRITTEN
```

```
10433                                                   ;ON DISK IS IN ERROR.
10434
10435                                                   ;GOOD DATA IS WHAT SHOULD BE ON DISK
10436                                                   ;BAD DATA IS WHAT IS GOING ON DISK
10437                                                   ;WORD NO IS 5.
10438  052360  000240                 ERHDGP: NOP
10439                                                   ;IF "ERROR 6" INSERTED BY
10440                                                   ;WRHEAD SUBROUTINE THEN HEADER
10441                                                   ;GAP GOING ON DISK IS WRONG.
10442
10443                                                   ;WORD NO. GIVES WHICH OF
10444                                                   ;THE HEADER GAP WORDS
10445                                                   ;ARE WRONG. FOR EXAMPLE:
10446
10447                                                   ;WORD NO 1 = FIRST HEADER
10448                                                   ;            GAP WORD
10449                                                   ;BAD WORD IS WHAT IS GOING ON DISK
10450
10451  052362  000240                 HDESYN: NOP
10452                                                   ;IF "ERROR 6" INSERTED BY
10453                                                   ;WRHEAD SUBROUTINE THEN LAST
10454                                                   ;HEADER GAP BYTE OR HEADER
10455                                                   ;SYNC BYTE GOING ON DISK IS WRONG.
10456
10457                                                   ;WORD NO = 5
10458                                                   ;BAD DATA IS WHAT IS GOING
10459                                                   ;ON DISK, RIGHT BYTE IS HEADER
10460                                                   ;GAP 0 BYTE, LEFT BYTE IS HEADER
10461                                                   ;GAP SYNC.
10462
10463
10464  052364  005737  002006                 TST     @#ERFLG$        ;ARE ANY ERRORS DETECTED
10465  052370  001004                         BNE     FOUT            ;IF YES EXIT ---------------------->
10466
10467  052372  004137  050566                 JSR     R1,@#WRDATA     ;WRITE THE DATA <---------------->
10468  052376  000000              FNWORD: .WORD   0               ;FORMAT COMMAND NO. OF DATA
10469  052400  000000                         .WORD   0
10470
10471  052402                      FOUT:
10472  052402  012605                         MOV     (SP)+,R5        ;;POP STACK INTO R5
10473  052404  012604                         MOV     (SP)+,R4        ;;POP STACK INTO R4
10474  052406  012603                         MOV     (SP)+,R3        ;;POP STACK INTO R3
10475  052410  012602                         MOV     (SP)+,R2        ;;POP STACK INTO R2
10476  052412  012601                         MOV     (SP)+,R1        ;;POP STACK INTO R1
10477  052414  012600                         MOV     (SP)+,R0        ;;POP STACK INTO R0
10478
10479  052416  000207                         RTS     PC
```

```
10480
10481                                   ;;*****************************************************************
10482                                   ;*WRITE HEADER
10483                                   ;;*****************************************************************
10484
10485
10486
10487
10488
10489                                   ;*R0 = MAINT.REG.
10490                                   ;*R1 = SIMULATED DISK
10491                                   ;*R2 = BYTE COUNT
10492                                   ;*R3 = WRITE WORD
10493                                   ;*R5 = WORD COUNT
10494
10495
10496
10497  052420  000000                   SCYL:   0                       ;
10498  052422  000000                   SSECTR: 0                       ;
10499  052424  000000                   SKEY1:  0                       ;
10500  052426  000000                   SKEY2:  0                       ;
10501  052430  000000                   SCRC:   0                       ;
10502
10503
10504  052432  012137  052420           WRHEAD: MOV     (R1)+,@#SCYL
10505  052436  012137  052422                   MOV     (R1)+,@#SSECTR
10506  052442  012137  052424                   MOV     (R1)+,@#SKEY1
10507  052446  012137  052426                   MOV     (R1)+,@#SKEY2
10508  052452  012137  052430                   MOV     (R1)+,@#SCRC
10509  052456  010146                           MOV     R1,-(SP)        ;;PUSH R1 ON STACK
10510  052460  012701  051020                   MOV     #SECGAP,R1      ;SIMULATED DISK INDICATOR
10511  052464  013700  001660                   MOV     @#RHMR,R0       ;R0 NOW HAS MAINT. REG. ADDR.
10512  052470  012710  000001                   MOV     #DMD,@R0        ;SET DIAG. MODE IN RHMR
10513  052474  012705  000002                   MOV     #2,R5           ;WORD COUNTER
10514  052500  052710  000010                   BIS     #MSTCK,@R0      ;SET SECTOR FOR FIRST BYTE
10515  052504  012710  000013           1$:     MOV     #MSTCK!MCLK!DMD,@R0;SET SECTOR, CLOCK, DIAG. MODE, RESET INDEX
10516  052510  032710  000040                   BIT     #MWR,@R0        ;CHECK WRITE BIT IN MAINT. REG.
10517  052514  001403                           BEQ     2$
10518  052516  000261                           SEC                     ;SET CARRY
10519  052520  006003                           ROR     R3              ;MOVE ONE FORWARD
10520  052522  000402                           BR      3$              ;
10521  052524  000241           2$:     CLC                             ;CLEAR CARRY
10522  052526  006003                           ROR     R3              ;MOVE ZERO FORWARD
10523  052530  012710  000001           3$:     MOV     #DMD,@R0        ;CLEAR CLOCK, SECTOR
10524  052534  012702  000007                   MOV     #7,R2           ;BYTE COUNTER
10525  052540  052710  000002           4$:     BIS     #MCLK,@R0       ;SET CLOCK
10526  052544  032710  000040                   BIT     #MWR,@R0        ;CHECK WRITE BIT IN MAINT.REG.
10527  052550  001403                           BEQ     5$              ;BRANCH IF ZERO
10528  052552  000261                           SEC                     ;SET CARRY
10529  052554  006003                           ROR     R3              ;MOVE ONE FORWARD
10530  052556  000402                           BR      6$
10531  052560  000241           5$:     CLC
10532  052562  006003                           ROR     R3
10533  052564  012710  000001           6$:     MOV     #DMD,@R0        ;SET DIAG. MODE AGAIN IN RHMR
10534  052570  005302                           DEC     R2
10535  052572  001362                           BNE     4$
```

```
10536  052574  005305                          DEC     R5
10537  052576  001342                          BNE     1$              ;CONTINUE
10538
10539  052600  010321                          MOV     R3,(R1)+
10540  052602  005703                          TST     R3
10541  052604  001414                          BEQ     7$
10542  052606  012737  000001  047320          MOV     #1,@#ERWORD
10543  052614  005037  001124                  CLR     @#$GDDAT
10544  052620  010337  001126                  MOV     R3,@#$BDDAT
10545  052624  012737  104006  052350          MOV     #104006,@#SEGPER
10546  052632  000137  053240                  JMP     @#17$           ;BRANCH OUT ---------------------->
10547
10548  052636  012702  000022          7$:     MOV     #18.,R2         ;COUNT NO. OF SECTOR GAP
10549  052642  012737  000024  047320  10$:     MOV     #20.,@#ERWORD   ;COUNT TO GIVE ERROR WORD
10550  052650  004737  050346                  JSR     PC,@#WRITE      ;WRITE SECTOR GAP
10551  052654  013721  050344                  MOV     @#WWORD,(R1)+   ;STORE SECTOR GAP WORD
10552  052660  001413                          BEQ     11$
10553  052662  160237  047320                  SUB     R2,@#ERWORD     ;IF NOT GET ERROR WORD NO.
10554  052666  005037  001124                  CLR     @#$GDDAT        ;GOOD WORD
10555  052672  013737  050344  001126          MOV     @#WWORD,@#$BDDAT;BAD WORD
10556  052700  012737  104006  052350          MOV     #104006,@#SEGPER;STORE "ERROR 6" IN SEGPER
10557  052706  000554                          BR      17$             ;BRANCH OUT ---------------------->
10558
10559  052710  005302                  11$:     DEC     R2              ;HAVE 18 WORDS OF ZEROS BEEN WRITTEN ?
10560  052712  001353                          BNE     10$             ;IF NOT DO SO
10561
10562                                          ;*AT THIS POINT THE SECTOR FOUND FLOP SHOULD
10563                                          ;*BE HIGH.   SO THAT THE HEADER SYNC BYTE CAN BE GIVEN
10564
10565                                          ;*HOWEVER IN THE DRIVE TIMING ERROR TEST THE REST OF THE ROUTINE
10566                                          ;*IS ABORTED - HEADER SYNC BYTE IS NOT GIVEN
10567
10568  052714  005737  002026                  TST     @#TESDTE        ;IS THIS A DRIVE TIMING ERROR
10569  052720  001147                          BNE     17$             ;BRANCH OUT IF YES
10570  052722  004737  050346                  JSR     PC,@#WRITE      ;WRITE ONE SECTOR GAP 0 BYTE
10571                                                                  ;AND ONE SYNC. BYTE = 230
10572  052726  013711  050344                  MOV     @#WWORD,(R1)    ;SAVE 0 BYTE AND SYNC BYTE
10573  052732  023721  047302                  CMP     @#RSYNC,(R1)+   ;IF SYNC. BYTE RIGHT
10574  052736  001414                          BEQ     12$             ;IF YES BRANCH
10575  052740  012737  000024  047320          MOV     #20.,@#ERWORD   ;IF NOT GET READY FOR ERROR
10576  052746  013737  047302  001124          MOV     @#RSYNC,@#$GDDAT;GOOD WORD
10577  052754  014137  001126                  MOV     -(R1),@#$BDDAT  ;BAD WORD
10578  052760  012737  104006  052352          MOV     #104006,@#FSYNER;INSERT "ERROR 6" IN FSYNER
10579  052766  000524                          BR      17$             ;BRANCH OUT ---------------------------->
10580
10581  052770  012702  000004          12$:     MOV     #4,R2           ;FOUR HEADER WORDS
10582  052774  012703  052420                  MOV     #SCYL,R3        ;POINTER FOR HEADER TABLE
10583  053000  012737  000005  047320  13$:     MOV     #5,@#ERWORD     ;ERROR WORD NO SET
10584  053006  004737  050346                  JSR     PC,@#WRITE      ;WRITE 4 HEADER WORDS
10585  053012  013711  050344                  MOV     @#WWORD,(R1)    ;STORE WRITTEN WORD
10586  053016  022321                          CMP     (R3)+,(R1)+     ;IS IT RIGHT?
10587  053020  001412                          BEQ     14$             ;IF GOOD CONTINUE
10588                                                                  ;IF NOT GET READY FOR PRINT
10589  053022  160237  047320                  SUB     R2,@#ERWORD     ;WORD NO
10590  053026  014337  001124                  MOV     -(R3),@#$GDDAT  ;GOOD DATA
10591  053032  014137  001126                  MOV     -(R1),@#$BDDAT  ;BAD DATA
```

```
10592  053036  012737  104006  052354          MOV     #104006,@#ERHEAD;INSERT "ERROP 6"
10593  053044  000475                          BR      17$             ;BRANCH OUT ------------------------->
10594
10595  053046  005302                  14$:    DEC     R2              ;ARE 4 HEADER WORDS DONE?
10596  053050  001353                          BNE     13$             ;IF NOT DO THEM
10597  053052  004737  050346                  JSR     PC,@#WRITE      ;WRITE CRC
10598  053056  013711  050344                  MOV     @#WWORD,(R1)    ;STORE CRC
10599  053062  022137  052346                  CMP     (R1)+,@#GCRC    ;COMPARE GOOD CRC
10600  053066  001414                          BEQ     20$             ;BRANCH IF GOOD
10601  053070  014137  001126                  MOV     -(R1),@#$BDDATA ;BAD CRC WRITTEN
10602  053074  013737  052346  001124          MOV     @#GCRC,@#$GDDAT ;GOOD CRC
10603  053102  012737  000005  047320          MOV     #5,@#ERWORD     ;ERROR WORD NO
10604  053110  012737  104006  052356          MOV     #104006,@#ERCRC ;INSERT ERROR 6
10605  053116  000450                          BR      17$             ;EXIT ------------------------------->
10606
10607  053120  012702  000005          20$:    MOV     #5,R2           ;NO OF HEADER GAP
10608  053124  012737  000006  047320  15$:    MOV     #6,@#ERWORD     ;ERROR WORD NO SET
10609  053132  004737  050346                  JSR     PC,@#WRITE      ;WRITE HEADER GAP
10610  053136  013721  050344                  MOV     @#WWORD,(R1)+   ;STORE
10611  053142  001412                          BEQ     16$             ;IF GOOD BRANCH
10612  053144  160237  047320                  SUB     R2,@#ERWORD     ;ERROR WORD NO
10613  053150  005037  001124                  CLR     @#$GDDAT        ;GOOD DATA
10614  053154  014137  001126                  MOV     -(R1),@#$BDDAT  ;BAD DATA
10615  053160  012737  104006  052360          MOV     #104006,@#ERHDGP;STORE "ERROR 6"
10616  053166  000424                          BR      17$             ;BRANCH OUT ------------------------->
10617
10618  053170  005302                  16$:    DEC     R2              ;ARE 5 HEADER GAP ZEROS  DONE
10619  053172  001354                          BNE     15$             ;IF NOT BRANCH
10620  053174  004737  050346                  JSR     PC,@#WRITE
10621  053200  013711  050344                  MOV     @#WWORD,(R1)
10622  053204  023721  047302                  CMP     @#RSYNC,(R1)+
10623  053210  001413                          BEQ     17$             ;EXIT ------------------------------->
10624  053212  012737  000005  047320          MOV     #5,@#ERWORD
10625  053220  014137  001126                  MOV     -(R1),@#$BDDAT
10626  053224  013737  047302  001124          MOV     @#RSYNC,@#$GDDAT
10627  053232  012737  104006  052362          MOV     #104006,@#HDESYN
10628
10629  053240                          17$:
10630  053240  012601                          MOV     (SP)+,R1        ;;POP STACK INTO R1
10631
10632  053242  000201                          RTS     R1
10633
10634
```

```
10635
10636
10637
10638
10639
10640
10641                                   ;;*****************************************************************
10642                                   ;*SEARCH SECTOR
10643                                   ;;*****************************************************************
10644
10645
10646
10647
10648
10649
10650                                   ;*      R0=RHMR ADDRESS
10651                                   ;*      R1=PASSED ARGUMENT (SECTOR SEARCHED FOR)
10652                                   ;*      R2=CLOCK COUNT (PER BYTE)
10653                                   ;*      R3=SECTOR COUNTER FROM R1
10654                                   ;*      R5=BYTES PER WORD COUNT
10655                                   ;*BEBORE INDEX IS GIVEN TWO SECTOR CLOCKS ARE GIVEN TO RESET
10656                                   ;*SECTOR PULSE IN CASE IT IS SET
10657                                   ;*AT BEGINNING OF EACH SECTOR ONE SECTOR CLOCK HAS TO RISE
10658                                   ;*BEFORE CLOCK THEN EVERY EIGHT CLOCKS ONE SECTOR CLOCK IS
10659                                   ;*IDENTICAL WITH CLOCK
10660                                   ;*NUMBERING THE SECTOR CLOCKS AS FOLLOWS
10661                                   ;*THE SECTOR CLOCK UNDER INDEX - 0
10662                                   ;*THE NEXT - 1
10663                                   ;*THE NEXT  - 2
10664                                   ;*ETC.
10665                                   ;*THEN THE LAST SECTOR CLOCK IN ONE SECTOR HAS NUMBER - 608
10666                                   ;*THE NEXT SECTOR THEN HAS 608 SECTOR CLOCKS
10667                                   ;*THE NEXT SECTOR THEN HAS ANOTHER 608 SECTOR CLOCKS
10668                                   ;*AND SO ON
10669
10670
10671
10672  053244  000000                   SECTR:  0                       ;SECTOR SEARCHED FOR
10673
10674  053246  012137  053244           SEARCH: MOV     (R1)+,  @#SECTR ;SAVE SECTOR SEARCHED FOR
10675  053252  010046                           MOV     R0,-(SP)        ;;PUSH R0 ON STACK
10676  053254  010146                           MOV     R1,-(SP)        ;;PUSH R1 ON STACK
10677  053256  010246                           MOV     R2,-(SP)        ;;PUSH R2 ON STACK
10678  053260  010346                           MOV     R3,-(SP)        ;;PUSH R3 ON STACK
10679  053262  010446                           MOV     R4,-(SP)        ;;PUSH R4 ON STACK
10680  053264  010546                           MOV     R5,-(SP)        ;;PUSH R5 ON STACK
10681  053266  013700  001660                   MOV     @#RHMR, R0      ;NOW R0 HAS MAINTENANCE REG. ADR.
10682  053272  013703  053244                   MOV     @#SECTR,R3      ;SECTOR COUNTER
10683  053276  012710  000001                   MOV     #DMD,   @R0     ;SET DIAGNOSTIC MODE
10684  053302  052710  000010                   BIS     #MSTCK, @R0     ;SET SECTOR CLOCK
10685  053306  042710  000010                   BIC     #MSTCK, @R0     ;CLEAR SECTOR CLOCK
10686  053312  052710  000010                   BIS     #MSTCK, @R0     ;SET SECTOR CLOCK
10687  053316  042710  000010                   BIC     #MSTCK, @R0     ;CLEAR SECTOR CLOCK
10688                                           ;THE ABOVE TWO SECTOR CLOCKS ARE GIVEN FOR
10689                                           ;RESETING SECTOR PULSE
10690                                           ;IN CASE IT STARTS SET
```

```
10691  053322  052710  000014          BIS     #MINX!MSTCK,@R0 ;SET INDEX AND SECTOR CLOCK
10692  053326  012710  000001          MOV     #DMD,   @R0     ;RESET INDEX AND SECTOR CLOCK
10693  053332  005703                  TST     R3              ;IF SECTOR REQUIRED JUMP OUT
10694  053334  001461                  BEQ     7$              ;BRANCH OF SECTOR ZERO REQUIRED
10695                                  ;*NOW THE 304 WORDS WILL START
10696
10697
10698
10699                          ;*FOR FIRST BYTE SECTOR CLOCK WILL GO HIGH THEN CLOCK WILL GO HIGH
10700                          ;*BOTH WILL COME DOWN TOGETHER THEN SEVEN CLOCKS WILL BE GIVEN
10701                          ;*;FOR SECOND BYTE AND ALL OTHER BYTES TILL NEXT SECTOR SECTOR CLOCK
10702                          ;*WILL BE IDENTICAL WITH ONE CLOCK
10703
10704
10705                                  ;*ONE WORD ONLY
10706
10707  053336  012702  000010  1$:     MOV     #8.,    R2      ;BYTE COUNTER
10708  053342  012705  000002          MOV     #2,     R5      ;BYTES PER WORD
10709  053346  052710  000010          BIS     #MSTCK,@R0      ;SET SECTOR CLOCK
10710  053352  052710  000002          BIS     #MCLK,@R0       ;SET CLOCK
10711  053356  000402                  BR      3$              ;BRANCH TO CLEAR SECTOR AND CLOCK
10712  053360  052710  000012  2$:     BIS     #MSTCK!MCLK,@R0 ;SET SECTOR AND CLOCK
10713  053364  042710  000012  3$:     BIC     #MSTCK!MCLK,@R0 ;CLEAR SECTOR AND CLOCK
10714  053370  052710  000002  8$:     BIS     #MCLK,  @R0     ;SET CLOCK
10715  053374  042710  000002          BIC     #MCLK,  @R0     ;CLEAR CLOCK
10716  053400  005302                  DEC     R2              ;BYTE COUNTER
10717  053402  001372                  BNE     8$              ;BRANCH IF BYTE NOT COMPLETE
10718  053404  012702  000007          MOV     #7,     R2      ;SETUP FOR SECOND BYTE
10719  053410  005305                  DEC     R5              ;IS WORD COMPLETE?
10720  053412  001362                  BNE     2$              ;BRANCH IF NOT COMPLETE
10721                                  ;TO GIVE SECTOR CLOCK AND CLOCK
10722
10723
10724                                  ;*NOW 303 WORDS ARE LEFT AND ALL ARE IDENTICAL
10725
10726  053414  012701  000457          MOV     #303.,  R1      ;WORDS PER SECTOR COUNTER
10727  053420  012705  000002  4$:     MOV     #2,     R5      ;BYTES PER WORD COUNTER
10728  053424  012702  000007  5$:     MOV     #7,     R2      ;BYTE COUNTER (CLOCK COUNTER)
10729  053430  052710  000012          BIS     #MSTCK!MCLK,@R0 ;SET SECTOR CLOCK AND CLOCK
10730  053434  042710  000012          BIC     #MSTCK!MCLK,@R0 ;CLEAR SECTOR CLOCK AND CLOCK
10731  053440  052710  000002  6$:     BIS     #MCLK,  @R0     ;SET CLOCK
10732  053444  042710  000002          BIC     #MCLK,  @R0     ;RESET CLOCK
10733  053450  005302                  DEC     R2              ;IS BYTE COMPLETE?
10734  053452  001372                  BNE     6$              ;BRANCH IF NOT COMPLETE
10735  053454  005305                  DEC     R5              ;IS WORD COMPLETE?
10736  053456  001362                  BNE     5$              ;BRANCH IF NOT
10737  053460  005301                  DEC     R1              ;IS SECTOR COMPLETE
10738  053462  001356                  BNE     4$              ;BRANCH IF NOT
10739  053464  052710  000010          BIS     #MSTCK,@R0      ;SET SECTOR
10740  053470  042710  000010          BIC     #MSTCK,@R0      ;CLEAR SECTOR
10741  053474  005303                  DEC     R3              ;IS REQUIRED NO OF SECTORS COMPLETE
10742  053476  001317                  BNE     1$              ;BRANCH IF NOT
10743
10744  053500                  7$:
10745  053500  012605                  MOV     (SP)+,R5        ;;POP STACK INTO R5
10746  053502  012604                  MOV     (SP)+,R4        ;;POP STACK INTO R4
```

HY

```
10747  053504  012603                  MOV     (SP)+,R3        ;;POP STACK INTO R3
10748  053506  012602                  MOV     (SP)+,R2        ;;POP STACK INTO R2
10749  053510  012601                  MOV     (SP)+,R1        ;;POP STACK INTO R1
10750  053512  012600                  MOV     (SP)+,R0        ;;POP STACK INTO R0
10751  053514  000201                  RTS     R1
10752
10753
10754                          ;;*****************************************************************
10755                          ;*READ ONE SECTOR OF DATA
10756                          ;;*****************************************************************
10757
10758  053516  000000          RNO:    0                       ;NO. OF WORDS READ
10759  053520  000000          RCOM:   0                       ;EXTRA STORAGE
10760
10761
10762
10763  053522  012137  053516  REDATA: MOV     (R1)+,@#RNO     ;SAVE NO. OF WORDS ONLY FOR INFORMATION
10764  053526  012137  053520          MOV     (R1)+,@#RCOM    ;EXTRA WORD ONLY FOR INFORMATION
10765  053532  010146                  MOV     R1,-(SP)        ;;PUSH R1 ON STACK
10766  053534  005737  002024          TST     @#TSECC         ;IS THIS AN ECC TEST
10767  053540  001403                  BEQ     1$              ;BRANCH IF NO
10768  053542  012737  177777  045234  MOV     #-1,@#TSECCG    ;THESE BITS ARE TO GENERATE ECC
10769  053550  012702  000402  1$:     MOV     #258.,R2        ;256 WORDS PER SECTOR
10770                                                          ;PLUS 2 ECC WORDS
10771  053554  012703  051116          MOV     #DISK,R3        ;POINTS TO DISK SIMULATION
10772  053560  012337  050110  2$:     MOV     (R3)+,@#WORD    ;READY TO READ CONTENTS
10773  053564  004737  050114          JSR     PC,@#READ       ;READ
10774  053570  005302                  DEC     R2              ;IS 256 WORDS DONE?
10775  053572  001372                  BNE     2$              ;IF NOT BRANCH
10776  053574  005737  002024          TST     @#TSECC         ;IS THIS AN ECC TEST
10777  053600  001012                  BNE     4$              ;BRANCH OUT IF YES
10778  053602  005037  045234          CLR     @#TSECCG        ;NO MORE ECC BITS ARE TO BE GENERATED
10779  053606  012702  000017          MOV     #15.,R2         ;ONE DATA GAP, 14 TOLERANCE GAP
10780  053612  012337  050110  3$:     MOV     (R3)+,@#WORD    ;READY TO READ CONTENTS OF WORD
10781  053616  004737  050114          JSR     PC,@#READ       ;READ
10782  053622  005302                  DEC     R2              ;COUNT
10783  053624  001372                  BNE     3$              ;BRANCH IF 14 NOT DONE
10784  053626                  4$:
10785  053626  012601                  MOV     (SP)+,R1        ;;POP STACK INTO R1
10786  053630  000201                  RTS     R1              ;RETURN
10787
10788
```

```
10789
10790
10791  053632                          RPVECT:
10792  053632  104401  053640                  TYPE    ,65$            ;;TYPE ASCIZ STRING
10793  053636  000421                          BR      64$             ;;GET OVER THE ASCIZ
10794                                  ;;65$:  .ASCIZ  /UNEXPECTED RP04 INTERRUPT @ PC = /
10795  053702                          64$:
10796  053702  104402                          TYPOC                           ;TYPE FROM PC
10797  053704  012777  053632  125714          MOV     #RPVECT,@RPVEC          ;RESTORE TRAP RP04 VECTOR
10798  053712  000000                          HALT                            ;CHANGE TO CONTINUE
```

```
10799                                   .SBTTL  SYSMAC LIBRARY ROUTINES
10800
10801                                   .SBTTL  SCOPE HANDLER ROUTINE
10802
10803                                   ;;*********************************************************************
10804                                   ;*THIS ROUTINE CONTROLS THE LOOPING OF SUBTESTS. IT WILL INCREMENT
10805                                   ;*AND LOAD THE TEST NUMBER($TSTNM) INTO THE DISPLAY REG.(DISPLAY<7:0>)
10806                                   ;*AND LOAD THE ERROR FLAG ($ERFLG) INTO DISPLAY<15:08>
10807                                   ;*THE SWITCH OPTIONS PROVIDED BY THIS ROUTINE ARE:
10808                                   ;*SW14=1        LOOP ON TEST
10809                                   ;*SW11=1        INHIBIT ITERATIONS
10810                                   ;*SW09=1        LOOP ON ERROR
10811                                   ;*SW08=1        LOOP ON TEST IN SWR<7:0>
10812                                   ;*CALL
10813                                   ;*      SCOPE           ;;SCOPE=IOT
10814
10815  053714                           $SCOPE:
10816  053714  104407                          CKSWR                   ;;TEST  FOR CHANGE IN SOFT-SWR
10817  053716  005037  047314                  CLR     @#NOSYNC        ;CLEAR FLAG FOR HEADER ERROR COMMANDS
10818  053722  005037  002024                  CLR     @#TSECC         ;CLEAR FLAG FOR ECC TEST
10819                                                                   ;WHEN =177777 IT IS AN ECC TEST
10820                                                                   ;WHEN =0 IT IS NOT AN ECC TEST
10821
10822  053726  005037  045234                  CLR     @#TSECCG        ;EVEN IN AN ECC TEST EVERY CLOCK
10823                                                                   ;IS NOT TO GENERATE ECC
10824                                                                   ;IF =177777 GENERATE ECC
10825                                                                   ;IF =0 DO NOT GENERATE ECC
10826  053732  005037  002026                  CLR     @#TESDTE        ;DRIVE TIMING ERROR TEST
10827  053736                           1$:
10828  053736  032777  040000  125174  1$:     BIT     #BIT14,@SWR     ;;LOOP ON PRESENT TEST?
10829  053744  001111                          BNE     $OVER           ;;YES IF SW14=1
10830                                   ;#####START OF CODE FOR THE XOR TESTER#####
10831  053746  000416                   $XTSTR: BR      6$              ;;IF RUNNING ON THE "XOR" TESTER CHANGE
10832                                                                   ;;THIS INSTRUCTION TO A "NOP" (NOP=240)
10833  053750  013746  000004                  MOV     @#ERRVEC,-(SP)  ;;SAVE THE CONTENTS OF THE ERROR VECTOR
10834  053754  012737  053774  000004          MOV     #5$,@#ERRVEC    ;;SET FOR TIMEOUT
10835  053762  005737  177060                  TST     @#177060        ;;TIME OUT ON XOR?
10836  053766  012637  000004                  MOV     (SP)+,@#ERRVEC  ;;RESTORE THE ERROR VECTOR
10837  053772  000463                          BR      $SVLAD          ;;GO TO THE NEXT TEST
10838  053774  022626                   5$:     CMP     (SP)+,(SP)+     ;;CLEAR THE STACK AFTER A TIME OUT
10839  053776  012637  000004                  MOV     (SP)+,@#ERRVEC  ;;RESTORE THE ERROR VECTOR
10840  054002  000423                          BR      7$              ;;LOOP ON THE PRESENT TEST
10841  054004                           6$:;#####END OF CODE FOR THE XOR TESTER#####
10842  054004  032777  000400  125126          BIT     #BIT08,@SWR     ;;LOOP ON SPEC. TEST?
10843  054012  001404                          BEQ     2$              ;;BR IF NO
10844  054014  127737  125120  001102          CMPB    @SWR,$TSTNM     ;;ON THE RIGHT TEST?    SWR<7:0>
10845  054022  001462                          BEQ     $OVER           ;;BR IF YES
10846  054024  105737  001103           2$:    TSTB    $ERFLG          ;;HAS AN ERROR OCCURRED?
10847  054030  001421                          BEQ     3$              ;;BR IF NO
10848  054032  123737  001115  001103          CMPB    $ERMAX,$ERFLG   ;;MAX. ERRORS FOR THIS TEST OCCURRED?
10849  054040  101015                          BHI     3$              ;;BR IF NO
10850  054042  032777  001000  125070          BIT     #BIT09,@SWR     ;;LOOP ON ERROR?
10851  054050  001404                          BEQ     4$              ;;BR IF NO
10852  054052  013737  001110  001106  7$:     MOV     $LPERR,$LPADR   ;;SET LOOP ADDRESS TO LAST SCOPE
10853  054060  000443                          BR      $OVER
10854  054062  105037  001103           4$:    CLRB    $ERFLG          ;;ZERO THE ERROR FLAG
```

L4

```
10855  054066  005037  001212                  CLR     $TIMES          ;;CLEAR THE NUMBER OF ITERATIONS TO MAKE
10856  054072  000415                          BR      1$              ;;ESCAPE TO THE NEXT TEST
10857  054074  032777  004000  125036  3$:     BIT     #BIT11,@SWR     ;;INHIBIT ITERATIONS?
10858  054102  001011                          BNE     1$              ;;BR IF YES
10859  054104  005737  001100                  TST     $PASS           ;;IF FIRST PASS OF PROGRAM
10860  054110  001406                          BEQ     1$              ;;      INHIBIT ITERATIONS
10861  054112  005237  001104                  INC     $ICNT           ;;INCREMENT ITERATION COUNT
10862  054116  023737  001212  001104          CMP     $TIMES,$ICNT    ;;CHECK THE NUMBER OF ITERATIONS MADE
10863  054124  002021                          BGE     $OVER           ;;BR IF MORE ITERATION REQUIRED
10864  054126  012737  000001  001104  1$:     MOV     #1,$ICNT        ;;REINITIALIZE THE ITERATION COUNTER
10865  054134  013737  054204  001212          MOV     $MXCNT,$TIMES   ;;SET NUMBER OF ITERATIONS TO DO
10866  054142  105237  001102          $SVLAD: INCB    $TSTNM          ;;COUNT TEST NUMBERS
10867  054146  011637  001106                  MOV     (SP),$LPADR     ;;SAVE SCOPE LOOP ADDRESS
10868  054152  011637  001110                  MOV     (SP),$LPERR     ;;SAVE ERROR LOOP ADDRESS
10869  054156  005037  001214                  CLR     $ESCAPE         ;;CLEAR THE ESCAPE FROM ERROR ADDRESS
10870  054162  112737  000001  001115          MOVB    #1,$ERMAX       ;;ONLY ALLOW ONE(1) ERROR ON NEXT TEST
10871  054170  013777  001102  124744  $OVER:  MOV     $TSTNM,@DISPLAY ;;DISPLAY TEST NUMBER
10872  054176  013716  001106                  MOV     $LPADR,(SP)     ;;FUDGE RETURN ADDRESS
10873  054202  000002                          RTI                     ;;FIXES PS
10874  054204  000004          $MXCNT: 4                               ;;MAX. NUMBER OF ITERATIONS
```

```
10875                                   .SBTTL  CONVERT BINARY TO DECIMAL AND TYPE ROUTINE
10876
10877                                   ;;**********************************************************************
10878                                   ;*THIS ROUTINE IS USED TO CHANGE A 16-BIT BINARY NUMBER TO A 5-DIGIT
10879                                   ;*SIGNED DECIMAL (ASCII) NUMBER AND TYPE IT. DEPENDING ON WHETHER THE
10880                                   ;*NUMBER IS POSITIVE OR NEGATIVE A SPACE OR A MINUS SIGN WILL BE TYPED
10881                                   ;*BEFORE THE FIRST DIGIT OF THE NUMBER. LEADING ZEROS WILL ALWAYS BE
10882                                   ;*REPLACED WITH SPACES.
10883                                   ;*CALL:
10884                                   ;*      MOV     NUM,-(SP)       ;;PUT THE BINARY NUMBER ON THE STACK
10885                                   ;*      TYPDS                   ;;GO TO THE ROUTINE
10886
10887  054206                           $TYPDS:
10888  054206  010046                          MOV     R0,-(SP)        ;;PUSH R0 ON STACK
10889  054210  010146                          MOV     R1,-(SP)        ;;PUSH R1 ON STACK
10890  054212  010246                          MOV     R2,-(SP)        ;;PUSH R2 ON STACK
10891  054214  010346                          MOV     R3,-(SP)        ;;PUSH R3 ON STACK
10892  054216  010546                          MOV     R5,-(SP)        ;;PUSH R5 ON STACK
10893  054220  012746  020200                  MOV     #20200,-(SP)    ;;SET BLANK SWITCH AND SIGN
10894  054224  016605  000020                  MOV     20(SP),R5       ;;GET THE INPUT NUMBER
10895  054230  100004                          BPL     1$              ;;BR IF INPUT IS POS.
10896  054232  005405                          NEG     R5              ;;MAKE THE BINARY NUMBER POS.
10897  054234  112766  000055  000001          MOVB    #'-,1(SP)       ;;MAKE THE ASCII NUMBER NEG.
10898  054242  005000                  1$:     CLR     R0              ;;ZERO THE CONSTANTS INDEX
10899  054244  012703  054422                  MOV     #$DBLK,R3       ;;SETUP THE OUTPUT POINTER
10900  054250  112723  000040                  MOVB    #' ,(R3)+       ;;SET THE FIRST CHARACTER TO A BLANK
10901  054254  005002                  2$:     CLR     R2              ;;CLEAR THE BCD NUMBER
10902  054256  016001  054412                  MOV     $DTBL(R0),R1    ;;GET THE CONSTANT
10903  054262  160105                  3$:     SUB     R1,R5           ;;FORM THIS BCD DIGIT
10904  054264  002402                          BLT     4$              ;;BR IF DONE
10905  054266  005202                          INC     R2              ;;INCREASE THE BCD DIGIT BY 1
10906  054270  000774                          BR      3$
10907  054272  060105                  4$:     ADD     R1,R5           ;;ADD BACK THE CONSTANT
10908  054274  005702                          TST     R2              ;;CHECK IF BCD DIGIT=0
10909  054276  001002                          BNE     5$              ;;FALL THROUGH IF 0
10910  054300  105716                          TSTB    (SP)            ;;STILL DOING LEADING 0'S?
10911  054302  100407                          BMI     7$              ;;BR IF YES
10912  054304  106316                  5$:     ASLB    (SP)            ;;MSD?
10913  054306  103003                          BCC     6$              ;;BR IF NO
10914  054310  116663  000001  177777          MOVB    1(SP),-1(R3)    ;;YES--SET THE SIGN
10915  054316  052702  000060          6$:     BIS     #'0,R2          ;;MAKE THE BCD DIGIT ASCII
10916  054322  052702  000040          7$:     BIS     #' ,R2          ;;MAKE IT A SPACE IF NOT ALREADY A DIGIT
10917  054326  110223                          MOVB    R2,(R3)+        ;;PUT THIS CHARACTER IN THE OUTPUT BUFFER
10918  054330  005720                          TST     (R0)+           ;;JUST INCREMENTING
10919  054332  020027  000010                  CMP     R0,#10          ;;CHECK THE TABLE INDEX
10920  054336  002746                          BLT     2$              ;;GO DO THE NEXT DIGIT
10921  054340  003002                          BGT     8$              ;;GO TO EXIT
10922  054342  010502                          MOV     R5,R2           ;;GET THE LSD
10923  054344  000764                          BR      6$              ;;GO CHANGE TO ASCII
10924  054346  105726                  8$:     TSTB    (SP)+           ;;WAS THE LSD THE FIRST NON-ZERO?
10925  054350  100003                          BPL     9$              ;;BP IF NO
10926  054352  116663  177777  177776          MOVB    -1(SP),-2(R3)   ;;YES--SET THE SIGN FOR TYPING
10927  054360  105013                  9$:     CLRB    (R3)            ;;SET THE TERMINATOR
10928  054362  012605                          MOV     (SP)+,R5        ;;POP STACK INTO R5
10929  054364  012603                          MOV     (SP)+,R3        ;;POP STACK INTO R3
10930  054366  012602                          MOV     (SP)+,R2        ;;POP STACK INTO R2
```

MY

```
 10931  054370  012601                          MOV     (SP)+,R1        ;;POP STACK INTO R1
 10932  054372  012600                          MOV     (SP)+,R0        ;;POP STACK INTO R0
 10933  054374  104401  054422                  TYPE    ,$DBLK          ;;NOW TYPE THE NUMBER
 10934  054400  016666  000002  000004          MOV     2(SP),4(SP)     ;;ADJUST THE STACK
 10935  054406  012616                          MOV     (SP)+,(SP)
 10936  054410  000002                          RTI                     ;;RETURN TO USER
 10937  054412  023420                  $DTBL:  10000.
 10938  054414  001750                          1000.
 10939  054416  000144                          100.
 10940  054420  000012                          10.
 10941  054422  000004                  $DBLK:  .BLKW   4
```

```
10942                                   .SBTTL  TYPE ROUTINE
10943
10944                                   ;;*****************************************************************
10945                                   ;*ROUTINE TO TYPE ASCIZ MESSAGE. MESSAGE MUST TERMINATE WITH A 0 BYTE.
10946                                   ;*THE ROUTINE WILL INSERT A NUMBER OF NULL CHARACTERS AFTER A LINE FEED.
10947                                   ;*NOTE1:        $NULL CONTAINS THE CHARACTER TO BE USED AS THE FILLER CHARACTER.
10948                                   ;*NOTE2:        $FILLS CONTAINS THE NUMBER OF FILLER CHARACTERS REQUIRED.
10949                                   ;*NOTE3:        $FILLC CONTAINS THE CHARACTER TO FILL AFTER.
10950                                   ;*
10951                                   ;*CALL:
10952                                   ;*1) USING A TRAP INSTRUCTION
10953                                   ;*      TYPE    ,MESADR         ;;MESADR IS FIRST ADDRESS OF AN ASCIZ STRING
10954                                   ;*OR
10955                                   ;*      TYPE
10956                                   ;*      MESADR
10957                                   ;*
10958
10959  054432  105737  001157          $TYPE:  TSTB    $TPFLG          ;;IS THERE A TERMINAL?
10960  054436  100002                          BPL     1$              ;;BR IF YES
10961  054440  000000                          HALT                    ;;HALT HERE IF NO TERMINAL
10962  054442  000407                          BR      3$              ;;LEAVE
10963  054444  010046                  1$:     MOV     R0,-(SP)        ;;SAVE R0
10964  054446  017600  000002                  MOV     @2(SP),R0       ;;GET ADDRESS OF ASCIZ STRING
10965  054452  112046                  2$:     MOVB    (R0)+,-(SP)     ;;PUSH CHARACTER TO BE TYPED ONTO STACK
10966  054454  001005                          BNE     4$              ;;BR IF IT ISN'T THE TERMINATOR
10967  054456  005726                          TST     (SP)+           ;;IF TERMINATOR POP IT OFF THE STACK
10968  054460  012600                  60$:    MOV     (SP)+,R0        ;;RESTORE R0
10969  054462  062716  000002          3$:     ADD     #2,(SP)         ;;ADJUST RETURN PC
10970  054466  000002                          RTI                     ;;RETURN
10971  054470  122716  000011          4$:     CMPB    #HT,(SP)        ;;BRANCH IF <HT>
10972  054474  001430                          BEQ     8$
10973  054476  122716  000200                  CMPB    #CRLF,(SP)      ;;BRANCH IF NOT <CRLF>
10974  054502  001006                          BNE     5$
10975  054504  005726                          TST     (SP)+           ;;POP  <CR><LF> EQUIV
10976  054506  104401                          TYPE                    ;;TYPE A CR AND LF
10977  054510  001223                          $CRLF
10978  054512  105037  054646                  CLRB    $CHARCNT        ;;CLEAR CHARACTER COUNT
10979  054516  000755                          BR      2$              ;;GET NEXT CHARACTER
10980  054520  004737  054602          5$:     JSR     PC,$TYPEC       ;;GO TYPE THIS CHARACTER
10981  054524  123726  001156          6$:     CMPB    $FILLC,(SP)+    ;;IS IT TIME FOR FILLER CHARS.?
10982  054530  001350                          BNE     2$              ;;IF NO GO GET NEXT CHAR.
10983  054532  013746  001154                  MOV     $NULL,-(SP)     ;;GET # OF FILLER CHARS. NEEDED
10984                                          ;;AND THE NULL CHAR.
10985  054536  105366  000001          7$:     DECB    1(SP)           ;;DOES A NULL NEED TO BE TYPED?
10986  054542  002770                          BLT     6$              ;;BR IF NO--GO POP THE NULL OFF OF STACK
10987  054544  004737  054602                  JSR     PC,$TYPEC       ;;GO TYPE A NULL
10988  054550  105337  054646                  DECB    $CHARCNT        ;;DO NOT COUNT AS A COUNT
10989  054554  000770                          BR      7$              ;;LOOP
10990
10991                                   ;HORIZONTAL TAB PROCESSOR
10992
10993  054556  112716  000040          8$:     MOVB    #' ,(SP)        ;;REPLACE TAB WITH SPACE
10994  054562  004737  054602          9$:     JSR     PC,$TYPEC       ;;TYPE A SPACE
10995  054566  132737  000007  054646          BITB    #7,$CHARCNT     ;;BRANCH IF NOT AT
10996  054574  001372                          BNE     9$              ;;TAB STOP
10997  054576  005726                          TST     (SP)+           ;;POP SPACE OFF STACK
```

```
10998  054600  000724                          BR      2$              ;;GET NEXT CHARACTER
10999  054602  105777  124342          $TYPEC: TSTB    @$TPS           ;;WAIT UNTIL PRINTER IS READY
11000  054606  100375                          BPL     $TYPEC
11001  054610  116677  000002  124334          MOVB    2(SP),@$TPB     ;;LOAD CHAR TO BE TYPED INTO DATA REG.
11002  054616  122766  000015  000002          CMPB    #CR,2(SP)       ;;IS CHARACTER A CARRIAGE RETURN?
11003  054624  001003                          BNE     1$              ;;BRANCH IF NO
11004  054626  105037  054646                  CLRB    $CHARCNT        ;;YES--CLEAR CHARACTER COUNT
11005  054632  000406                          BR      $TYPEX          ;;EXIT
11006  054634  122766  000012  000002  1$:     CMPB    #LF,2(SP)       ;;IS CHARACTER A LINE FEED?
11007  054642  001402                          BEQ     $TYPEX          ;;BRANCH IF YES
11008  054644  105227                          INCB    (PC)+           ;;COUNT THE CHARACTER
11009  054646  000000          $CHARCNT:.WORD  0               ;;CHARACTER COUNT STORAGE
11010  054650  000207          $TYPEX: RTS     PC
11011
```

```
                        .SBTTL  TTY INPUT ROUTINE

                        ;;*******************************************************************
                        .ENABL  LSB
054652  000000          $TKCNT: .WORD   0               ;;NUMBER OF ITEMS IN QUEUE
054654  000000          $TKQIN: .WORD   0               ;;INPUT POINTER
054656  000000          $TKQOUT: .WORD  0               ;;OUTPUT POINTER
054660  000011          $TKQSRT: .BLKB  9.              ;;TTY KEYBOARD QUEUE
        054671          $TKQEND=.
        054672          .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
                        ;
054672  005037  054652  $TKINT: CLR      $TKCNT          ;;CLEAR COUNT OF ITEMS IN QUEUE
054676  012737  054660  054654      MOV     #$TKQSRT,$TKQIN ;;MOVE THE STARTING ADDRESS OF THE
054704  013737  054654  054656      MOV     $TKQIN,$TKQOUT  ;;QUEUE INTO THE INPUT & OUTPUT POINTERS.
054712  012737  054742  000060      MOV     #$TKSRV,@#TKVEC ;;INITIALIZE THE KEYBOARD VECTOR
054720  012737  000200  000062      MOV     #200,@#TKVEC+2  ;;"BR" LEVEL 4
054726  005777  124214              TST     @$TKB           ;;CLEAR DONE FLAG
054732  012777  000100  124204      MOV     #100,@$TKS      ;;ENABLE TTY KEYBOARD INTERRUPT
054740  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)
                        ;
054742  117746  124200  $TKSRV: MOVB     @$TKB,-(SP)     ;;PICKUP THE CHARACTER
054746  042716  177600          BIC     #^C177,(SP)     ;;STRIP THE JUNK
054752  021627  000003          CMP     (SP),#3         ;;IS IT A CONTROL C?
054756  001007                  BNE     1$              ;;BRANCH IF NO
054760  104401  055731          TYPE    ,$CNTLC         ;;TYPE A CONTROL-C (^C)
054764  004737  054672          JSR     PC,$TKINT       ;;INIT THE KEYBOARD
054770  005726                  TST     (SP)+           ;;CLEAN UP STACK
054772  000137  041360          JMP     OPERSEL         ;;CONTROL C RESTART
054776  021627  000007  1$:     CMP     (SP),#7         ;;IS IT A CONTROL G?
055002  001004                  BNE     2$              ;;BRANCH IF NO
055004  022737  000176  001140      CMP     #SWREG,SWR      ;;IS SOFT-SWR SELECTED?
055012  001500                  BEQ     6$              ;;GO TO SWR CHANGE

055014                  2$:
055014  022737  000011  054652      CMP     #9.,$TKCNT      ;;IS THE QUEUE FULL?
055022  001004                  BNE     3$              ;;BRANCH IF NO
055024  104401  001216          TYPE    ,$BELL          ;;RING THE TTY BELL
055030  005726                  TST     (SP)+           ;;CLEAN CHARACTER OFF OF STACK
055032  000451                  BR      5$              ;;EXIT
055034  021627  000023  3$:     CMP     (SP),#23        ;;IS IT A CONTROL-S?
055040  001021                  BNE     32$             ;;BRANCH IF NO
```

```
11068  055042  005077  124076                  CLR     @$TKS           ;;DISABLE TTY KEYBOARD INTERRUPTS
11069  055046  005726                          TST     (SP)+           ;;CLEAN CHAR OFF STACK
11070  055050  105777  124070          31$:    TSTB    @$TKS           ;;WAIT FOR A CHAR
11071  055054  100375                          BPL     31$             ;;LOOP UNTIL ITS THERE
11072  055056  117746  124064                  MOVB    @$TKB,-(SP)     ;;GET THE CHARACTER
11073  055062  042716  177600                  BIC     #^C177,(SP)     ;;MAKE IT 7-BIT ASCII
11074  055066  022627  000021                  CMP     (SP)+,#21       ;;IS IT A CONTROL-Q?
11075  055072  001366                          BNE     31$             ;;BRANCH IF NO
11076  055074  012777  000100  124042          MOV     #100,@$TKS      ;;REENABLE TTY KEYBOARD INTERRUPTS
11077  055102  000002                          RTI                     ;;RETURN
11078  055104  005237  054652          32$:    INC     $TKCNT          ;;COUNT THIS CHARACTER
11079  055110  021627  000140                  CMP     (SP),#140       ;;IS IT UPPER CASE?
11080  055114  002405                          BLT     4$              ;;BRANCH IF YES
11081  055116  021627  000175                  CMP     (SP),#175       ;;IS IT A SPECIAL CHAR?
11082  055122  003002                          BGT     4$              ;;BRANCH IF YES
11083  055124  042716  000040                  BIC     #40,(SP)        ;;MAKE IT UPPER CASE
11084  055130  112677  177520          4$:     MOVB    (SP)+,@$TKQIN   ;;AND PUT IT IN QUEUE
11085  055134  005237  054654                  INC     $TKQIN          ;;UPDATE THE POINTER
11086  055140  023727  054654  054671          CMP     $TKQIN,#$TKQEND ;;GO OFF THE END?
11087  055146  001003                          BNE     5$              ;;BRANCH IF NO
11088  055150  012737  054660  054654          MOV     #$TKQSRT,$TKQIN ;;RESET THE POINTER
11089  055156  000002                  5$:     RTI                     ;;RETURN
11090
11091                                  ;;**********************************************************************
11092                                  ;*SOFTWARE SWITCH REGISTER CHANGE ROUTINE.
11093                                  ;*ROUTINE IS ENTERED FROM THE TRAP HANDLER, AND WILL
11094                                  ;*SERVICE THE TEST FOR CHANGE IN SOFTWARE SWITCH REGISTER TRAP
11095                                  ;*CALL WHEN OPERATING IN TTY INTERRUPT MODE.
11096  055160  022737  000176  001140  $CKSWR: CMP     #SWREG,SWP      ;;IS THE SOFT-SWR SELECTED
11097  055166  001124                          BNE     15$             ;;EXIT IF NOT
11098  055170  105777  123750                  TSTB    @$TKS           ;;IS A CHAR WAITING?
11099  055174  100121                          BPL     15$             ;;IF NOT, EXIT
11100  055176  117746  123744                  MOVB    @$TKB,-(SP)     ;;YES
11101  055202  042716  177600                  BIC     #^C177,(SP)     ;;MAKE IT  7-BIT ASCII
11102  055206  021627  000007                  CMP     (SP),#7         ;;IS IT A CONTROL-G?
11103  055212  001300                          BNE     2$              ;;IF NOT, PUT IT IN THE TTY QUEUE
11104                                                                  ;;AND EXIT
11105
11106                                  ;;**********************************************************************
11107                                  ;*CONTROL IS PASSED TO THIS POINT FROM EITHER THE TTY INTERRUPT SERVICE
11108                                  ;*ROUTINE OR FROM THE SOFTWARE SWITCH REGISTER TRAP CALL, AS A RESULT OF A
11109                                  ;*CONTROL-G BEING TYPED, AND THE SOFTWARE SWITCH REGISTER BEING SELECTED.
11110  055214  123727  001134  000001  6$:     CMPB    $AUTOB,#1       ;;ARE WE RUNNING IN AUTO-MODE?
11111  055222  001674                          BEQ     2$              ;;BRANCH IF YES
11112  055224  005726                          TST     (SP)+           ;;CLEAR CONTROL-G OFF STACK
11113  055226  004737  054672                  JSR     PC,$TKINT       ;;FLUSH THE TTY INPUT QUEUE
11114  055232  005077  123706                  CLR     @$TKS           ;;DISABLE TTY KEYBOARD INTERRUPTS
11115  055236  112737  000001  001135          MOVB    #1,$INTAG       ;;SET INTERRUPT MODE INDICATOR
11116
11117  055244  104401  055743                  TYPE    ,$CNTLG         ;;ECHO THE CONTROL-G (^G)
11118  055250  104401  055750          $GTSWR: TYPE    ,$MSWR          ;;TYPE CURRENT CONTENTS
11119  055254  013746  000176                  MOV     SWREG,-(SP)     ;;SAVE SWREG FOR TYPEOUT
11120  055260  104402                          TYPOC                   ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
11121  055262  104401  055761                  TYPE    ,$MNEW          ;;PROMPT FOR NEW SWR
11122  055266  005046                  19$:    CLR     -(SP)           ;;CLEAR COUNTER
11123  055270  005046                          CLR     -(SP)           ;;THE NEW SWR
```

```
11124  055272  105777  123646          7$:     TSTB    @$TKS           ;;CHAR THERE?
11125  055276  100375                          BPL     7$              ;;IF NOT TRY AGAIN
11126
11127  055300  117746  123642                  MOVP    @$TKB,-(SP)     ;;PICK UP CHAR
11128  055304  042716  177600                  BIC     #^C177,(SP)     ;;MAKE IT 7-BIT ASCII
11129
11130  055310  021627  000003                  CMP     (SP),#3         ;;IS IT A CONTROL-C?
11131  055314  001015                          BNE     9$              ;;BRANCH IF NOT
11132  055316  104401  055731                  TYPE    ,$CNTLC         ;;YES, ECHO CONTROL-C (^C)
11133  055322  062706  000006                  ADD     #6,SP           ;;CLEAN UP STACK
11134  055326  123727  001135  000001          CMPB    $INTAG,#1       ;;REENABLE TTY KEYBOARD INTERRUPTS?
11135  055334  001003                          BNE     8$              ;;BRANCH IF NO
11136  055336  012777  000100  123600          MOV     #100,@$TKS      ;;ALLOW TTY KEYBOARD INTERRUPTS
11137  055344  000137  041360          8$:     JMP     OPERSEL         ;;CONTROL-C RESTART
11138
11139
11140  055350  021627  000025          9$:     CMP     (SP),#25        ;;IS IT A CONTROL-U?
11141  055354  001005                          BNE     10$             ;;BRANCH IF NOT
11142  055356  104401  055736                  TYPE    ,$CNTLU         ;;YES, ECHO CONTROL-U (^U)
11143  055362  062706  000006          20$:    ADD     #6,SP           ;;IGNORE PREVIOUS INPUT
11144  055366  000737                          BR      19$             ;;LET'S TRY IT AGAIN
11145
11146
11147  055370  021627  000015          10$:    CMP     (SP),#15        ;;IS IT A <CR>?
11148  055374  001022                          BNE     16$             ;;BRANCH IF NO
11149  055376  005766  000004                  TST     4(SP)           ;;YES, IS IT THE FIRST CHAR?
11150  055402  001403                          BEQ     11$             ;;BRANCH IF YES
11151  055404  016677  000002  123526          MOV     2(SP),@SWR      ;;SAVE NEW SWR
11152  055412  062706  000006          11$:    ADD     #6,SP           ;;CLEAR UP STACK
11153  055416  104401  001223          14$:    TYPE    ,$CRLF          ;;ECHO <CR> AND <LF>
11154  055422  123727  001135  000001          CMPB    $INTAG,#1       ;;RE-ENABLE TTY KBD INTERRUPTS?
11155  055430  001003                          BNE     15$             ;;BRANCH IF NOT
11156  055432  012777  000100  123504          MOV     #100,@$TKS      ;;RE-ENABLE TTY KPD INTERRUPTS
11157  055440  000002          15$:    RTI                             ;;RETURN
11158  055442  004737  054602          16$:    JSR     PC,$TYPEC       ;;ECHO CHAR
11159  055446  021627  000060                  CMP     (SP),#60        ;;CHAR < 0?
11160  055452  002420                          BLT     18$             ;;BRANCH IF YES
11161  055454  021627  000067                  CMP     (SP),#67        ;;CHAR > 7?
11162  055460  003015                          BGT     18$             ;;BRANCH IF YES
11163  055462  042726  000060                  BIC     #60,(SP)+       ;;STRIP-OFF ASCII
11164  055466  005766  000002                  TST     2(SP)           ;;IS THIS THE FIRST CHAR
11165  055472  001403                          BEQ     17$             ;;BRANCH IF YES
11166  055474  006316                          ASL     (SP)            ;;NO, SHIFT PRESENT
11167  055476  006316                          ASL     (SP)            ;;   CHAR OVER TO MAKE
11168  055500  006316                          ASL     (SP)            ;;   ROOM FOR NEW ONE.
11169  055502  005266  000002          17$:    INC     2(SP)           ;;KEEP COUNT OF CHAR
11170  055506  056616  177776                  BIS     -2(SP),(SP)     ;;SET IN NEW CHAR
11171  055512  000667                          BR      7$              ;;GET THE NEXT ONE
11172  055514  104401  001222          18$:    TYPE    ,$QUES          ;;TYPE ?<CR><LF>
11173  055520  000720                          BR      20$             ;;SIMULATE CONTROL-U
11174                                  .DSABL  LSB
11175
11176
11177                                  ;;**************************************************************
11178                                  ;*THIS ROUTINE WILL INPUT A SINGLE CHARACTER FROM THE TTY
11179                                  ;*CALL:
```

```
11180                                ;*      RDCHR                   ;;GET A CHARACTER FROM THE QUEUE
11181                                ;*      RETURN HERE             ;;CHARACTER IS ON THE STACK
11182                                ;*                              ;;WITH PARITY BIT STRIPPED OFF
11183                                ;
11184
11185  055522  011646                $RDCHR: MOV     (SP),-(SP)      ;;PUSH DOWN THE PC AND
11186  055524  016666  000004  000002        MOV     4(SP),2(SP)     ;;THE PS
11187  055532  005066  000004                CLR     4(SP)           ;;GET READY FOR A CHARACTER
11188  055536  005046                        CLR     -(SP)           ;;PUT NEW PS ON STACK
11189  055540  012746  055546                MOV     #64$,-(SP)      ;;PUT NEW PC ON  STACK
11190  055544  000002                        RTI                     ;;POP NEW PC AND PS
11191  055546                        64$:
11192  055546  005737  054652        1$:     TST     $TKCNT          ;;WAIT ON A CHARACTER
11193  055552  001775                        BEQ     1$
11194  055554  005337  054652                DEC     $TKCNT          ;;DECREMENT THE COUNTER
11195  055560  117766  177072  000004        MOVB    @$TKQOUT,4(SP)  ;;GET ONE CHARACTER
11196  055566  005237  054656                INC     $TKQOUT         ;;UPDATE THE POINTER
11197  055572  023727  054656  054671        CMP     $TKQOUT,#$TKQEND ;;DID IT GO OFF OF THE END?
11198  055600  001003                        BNE     2$              ;;BRANCH IF NO
11199  055602  012737  054660  054656        MOV     #$TKQSRT,$TKQOUT ;;RESET THE POINTER
11200  055610  000002                2$:     RTI                     ;;RETURN
11201                                ;;**************************************************************
11202                                ;*THIS ROUTINE WILL INPUT A STRING FROM THE TTY
11203                                ;*CALL:
11204                                ;*      RDLIN                   ;;INPUT A STRING FROM THE TTY
11205                                ;*      RETURN HERE             ;;ADDRESS OF FIRST CHARACTER WILL BE ON THE STACK
11206                                ;*                              ;;TERMINATOR WILL BE A BYTE OF ALL 0'S
11207
11209  055612  010346                $RDLIN: MOV     R3,-(SP)        ;;SAVE R3
11209  055614  012703  055720        1$:     MOV     #$TTYIN,R3      ;;GET ADDRESS
11210  055620  022703  055731        2$:     CMP     #$TTYIN+9.,R3   ;;BUFFER FULL?
11211  055624  101405                        BLOS    4$              ;;BR IF YES
11212  055626  104410                        RDCHR                   ;;GO READ ONE CHARACTER FROM THE TTY
11213  055630  112613                        MOVB    (SP)+,(R3)      ;;GET CHARACTER
11214  055632  122713  000177        10$:    CMPB    #177,(R3)       ;;IS IT A RUBOUT
11215  055636  001003                        BNE     3$              ;;SKIP IF NOT
11216  055640  104401  001222        4$:     TYPE    ,$QUES          ;;TYPE A '?'
11217  055644  000763                        BR      1$              ;;CLEAR THE BUFFER AND LOOP
11218  055646  111337  055716        3$:     MOVB    (R3),9$         ;;ECHO THE CHARACTER
11219  055652  104401  055716                TYPE    ,9$
11220  055656  122723  000015                CMPB    #15,(P3)+       ;;CHECK FOR RETURN
11221  055662  001356                        BNE     2$              ;;LOOP IF NOT RETURN
11222  055664  105063  177777                CLRB    -1(R3)          ;;CLEAR RETURN (THE 15)
11223  055670  104401  001224                TYPE    ,SLF            ;;TYPE A LINE FEED
11224  055674  012603                        MOV     (SP)+,R3        ;;RESTORE R3
11225  055676  011646                        MOV     (SP),-(SP)      ;;ADJUST THE STACK AND PUT ADDRESS OF THE
11226  055700  016666  000004  000002        MOV     4(SP),2(SP)     ;;      FIRST ASCII CHARACTER ON IT
11227  055706  012766  055720  000004        MOV     #$TTYIN,4(SP)
11228  055714  000002                        RTI                     ;;RETURN
11229  055716     000                9$:     .BYTE   0               ;;STORAGE FOR ASCII CHAR. TO TYPE
11230  055717     000                        .BYTE   0               ;;TERMINATOR
11231  055720  000011                $TTYIN: .BLKB   9.              ;;RESERVE 9. BYTES FOR TTY INPUT
11232  055731     136  006503  000012  $CNTLC: .ASCIZ  /^C/<15><12>    ;;CONTROL "C"
11233  055736  052536  005015     000  $CNTLU: .ASCIZ  /^U/<15><12>    ;;CONTROL "U"
11234  055743     136  006507  000012  $CNTLG: .ASCIZ  /^G/<15><12>    ;;CONTROL "G"
11235  055750  005015  053523  020122  $MSWR:  .ASCIZ  <15><12>/SWR = /
```

```
11236  055756  020075     000
11237  055761     040  047040  053505  $MNEW:  .ASCIZ  /  NEW = /
11238  055766  036440  000040
11239                                                          ;FROM THE TTY
```

```
11240                                   .SBTTL  READ AN OCTAL NUMBER FROM THE TTY
11241
11242                                   ;;*******************************************************************
11243                                   ;*THIS ROUTINE WILL READ AN OCTAL (ASCII) NUMBER FROM THE TTY AND
11244                                   ;*CHANGE IT TO BINARY.
11245                                   ;*THE INPUT CHARACTERS WILL BE CHECKED TO INSURED THEY ARE LEGAL
11246                                   ;*OCTAL DIGITS. IF AN ILLEGAL CHARACTER IS READ A "?" WILL BE TYPED
11247                                   ;*FOLLOWED BY A CARRIAGE RETURN-LINE FEED. THE COMPLETE NUMBER MUST
11248                                   ;*THEN BE RETYPED. THE INPUT IS TERMINATED BY TYPING A CARRIAGE RETURN.
11249                                   ;*CALL:
11250                                   ;*      RDOCT                   ;;READ AN OCTAL NUMBER
11251                                   ;*      RETURN HERE             ;;LOW ORDER BITS ARE ON TOP OF THE STACK
11252                                   ;*                              ;;HIGH ORDER BITS ARE IN $HIOCT
11253
11254  055772  011646                  $RDOCT: MOV     (SP),-(SP)      ;;PROVIDE SPACE FOR THE
11255  055774  016666  000004  000002          MOV     4(SP),2(SP)     ;;INPUT NUMBER
11256  056002  010046                          MOV     R0,-(SP)        ;;PUSH R0 ON STACK
11257  056004  010146                          MOV     R1,-(SP)        ;;PUSH R1 ON STACK
11258  056006  010246                          MOV     R2,-(SP)        ;;PUSH R2 ON STACK
11259  056010  104411                  1$:     RDLIN                   ;;READ AN ASCIZ LINE
11260  056012  012600                          MOV     (SP)+,R0        ;;GET ADDRESS OF 1ST CHARACTER
11261  056014  010037  056120                  MOV     R0,5$           ;;AND SAVE IT
11262  056020  005001                          CLR     R1              ;;CLEAR DATA WORD
11263  056022  005002                          CLR     R2
11264  056024  112046                  2$:     MOVB    (R0)+,-(SP)     ;;PICKUP THIS CHARACTER
11265  056026  001420                          BEQ     3$              ;;IF ZERO GET OUT
11266  056030  122716  000060                  CMPB    #'0,(SP)        ;;MAKE SURE THIS CHARACTER
11267  056034  003026                          BGT     4$              ;;IS AN OCTAL DIGIT
11268  056036  122716  000067                  CMPB    #'7,(SP)
11269  056042  002423                          BLT     4$
11270  056044  006301                          ASL     R1              ;;*2
11271  056046  006102                          ROL     R2
11272  056050  006301                          ASL     R1              ;;*4
11273  056052  006102                          ROL     R2
11274  056054  006301                          ASL     R1              ;;*8
11275  056056  006102                          ROL     R2
11276  056060  042716  177770                  BIC     #^C7,(SP)       ;;STRIP THE ASCII JUNK
11277  056064  062601                          ADD     (SP)+,R1        ;;ADD IN THIS DIGIT
11278  056066  000756                          BR      2$              ;;LOOP
11279  056070  005726                  3$:     TST     (SP)+           ;;CLEAN TERMINATOR FROM STACK
11280  056072  010166  000012                  MOV     R1,12(SP)       ;;SAVE THE RESULT
11281  056076  010237  056130                  MOV     R2,$HIOCT
11282  056102  012602                          MOV     (SP)+,R2        ;;POP STACK INTO R2
11283  056104  012601                          MOV     (SP)+,R1        ;;POP STACK INTO R1
11284  056106  012600                          MOV     (SP)+,R0        ;;POP STACK INTO R0
11285  056110  000002                          RTI                     ;;RETURN
11286  056112  005726                  4$:     TST     (SP)+           ;;CLEAN PARTIAL FROM STACK
11287  056114  105010                          CLRB    (R0)            ;;SET A TERMINATOR
11288  056116  104401                          TYPE                    ;;TYPE UP THRU THE BAD CHAR.
11289  056120  000000                  5$:     .WORD   0
11290  056122  104401  001222                  TYPE    ,$QUES          ;;"?" "CR" & "LF"
11291  056126  000730                          BR      1$              ;;TRY AGAIN
11292  056130  000000                  $HIOCT: .WORD   0               ;;HIGH ORDER BITS GO HERE
```

```
11293                                   .SBTTL  ERROR HANDLER ROUTINE
11294
11295                                   ;;*****************************************************************
11296                                   ;*THIS ROUTINE WILL INCREMENT THE ERROR FLAG AND THE ERROR COUNT,
11297                                   ;*SAVE THE ERROR ITEM NUMBER AND THE ADDRESS OF THE ERROR CALL
11298                                   ;*AND GO TO $ERRTYP ON ERROR
11299                                   ;*THE SWITCH OPTIONS PROVIDED BY THIS ROUTINE ARE:
11300                                   ;*SW15=1        HALT ON ERROR
11301                                   ;*SW13=1        INHIBIT ERROR TYPEOUTS
11302                                   ;*SW10=1        BELL ON ERROR
11303                                   ;*SW09=1        LOOP ON ERROR
11304                                   ;*CALL
11305                                   ;*      ERROR   N       ;;ERROR=EMT AND N=ERROR ITEM NUMBER
11306
11307  056132                           $ERROP:
11308  056132  104407                          CKSWR                   ;;TEST FOR CHANGE IN SOFT-SWR
11309  056134  012737  177777  002006          MOV     #-1,@#ERFLG$    ;SET ERROR FLAG
11310  056142                           REGSA1:
11311  056142  105237  001103          7$:     INCB    $ERFLG          ;;SET THE ERROR FLAG
11312  056146  001775                          BEQ     7$              ;;DON'T LET THE FLAG GO TO ZERO
11313  056150  013777  001102  122764          MOV     $TSTNM,@DISPLAY ;;DISPLAY TEST NUMBER AND ERROR FLAG
11314  056156  032777  002000  122754          BIT     #BIT10,@SWR     ;;BELL ON ERROR?
11315  056164  001402                          BEQ     1$              ;;NO - SKIP
11316  056166  104401  001216                  TYPE    ,$BELL          ;;RING BELL
11317  056172  005237  001112          1$:     INC     $ERTTL          ;;COUNT THE NUMBER OF ERRORS
11318  056176  011637  001116                  MOV     (SP),$ERRPC     ;;GET ADDRESS OF ERROR INSTRUCTION
11319  056202  162737  000002  001116          SUB     #2,$ERRPC
11320  056210  117737  122702  001114          MOVB    @$ERRPC,$ITEMB  ;;STRIP AND SAVE THE ERROR ITEM CODE
11321  056216  032777  020000  122714          BIT     #BIT13,@SWR     ;;SKIP TYPEOUT IF SET
11322  056224  001004                          BNE     20$             ;;SKIP TYPEOUTS
11323  056226  004737  056300                  JSR     PC,$ERRTYP      ;;GO TO USER ERROR ROUTINE
11324  056232  104401  001223                  TYPE    ,$CRLF
11325  056236                           20$:
11326  056236  005777  122676          2$:     TST     @SWR            ;;HALT ON ERROR
11327  056242  100002                          BPL     3$              ;;SKIP IF CONTINUE
11328  056244  000000                          HALT                    ;;HALT ON ERROR!
11329  056246  104407                          CKSWR                   ;;TEST FOR CHANGE IN SOFT-SWR
11330  056250  032777  001000  122662  3$:     BIT     #BIT09,@SWR     ;;LOOP ON ERROR SWITCH SET?
11331  056256  001402                          BEQ     4$              ;;BR IF NO
11332  056260  013716  001110                  MOV     $LPERR,(SP)     ;;FUDGE RETURN FOR LOOPING
11333  056264  005737  001214          4$:     TST     $ESCAPE         ;;CHECK FOR AN ESCAPE ADDRESS
11334  056270  001402                          BEQ     5$              ;;BR IF NONE
11335  056272  013716  001214                  MOV     $ESCAPE,(SP)    ;;FUDGE RETURN ADDRESS FOR ESCAPE
11336  056276                           5$:
11337  056276  000002                          RTI                     ;;RETURN
```

```
                        .SBTTL  ERROR MESSAGE TYPEOUT ROUTINE

                        ;;**********************************************************************
                        ;*THIS ROUTINE USES THE "ITEM CONTROL BYTE" ($ITEMB) TO DETERMINE WHICH
                        ;*ERROR IS TO BE REPORTED. IT THEN OBTAINS, FROM THE "ERROR TABLE" ($ERRTB),
                        ;*AND REPORTS THE APPROPRIATE INFORMATION CONCERNING THE ERROR.

056300                  $ERRTYP:
056300  104401  001223          TYPE    ,$CRLF          ;;"CARRIAGE RETURN" & "LINE FEED"
056304  010046                  MOV     R0,-(SP)        ;;SAVE R0
056306  005000                  CLR     R0              ;;PICKUP THE ITEM INDEX
056310  153700  001114          BISB    @#$ITEMB,R0
056314  001004                  BNE     1$              ;;IF ITEM NUMBER IS ZERO, JUST
                                                        ;;TYPE THE PC OF THE ERROR
056316  013746  001116          MOV     $ERRPC,-(SP)    ;;SAVE $ERRPC FOR TYPEOUT
                                                        ;;ERROR ADDRESS
056322  104402                  TYPOC                   ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
056324  000445                  BR      10$             ;;GET OUT
056326  005300          1$:     DEC     R0              ;;ADJUST THE INDEX SO THAT IT WILL
056330  006300                  ASL     R0              ;;      WORK FOR THE ERROR TABLE
056332  006300                  ASL     R0
056334  006300                  ASL     R0
056336  062700  001226          ADD     #$ERRTB,R0      ;;FORM TABLE POINTER
056342  012037  056352          MOV     (R0)+,2$        ;;PICKUP "ERROR MESSAGE" POINTER
056346  001404                  BEQ     3$              ;;SKIP TYPEOUT IF NO POINTER
056350  104401                  TYPE                    ;;TYPE THE "ERROR MESSAGE"
056352  000000          2$:     .WORD   0               ;;"ERROR MESSAGE" POINTER GOES HERE
056354  104401  001223          TYPE    ,$CRLF          ;;"CARRIAGE RETURN" & "LINE FEED"
056360  012037  056370  3$:     MOV     (R0)+,4$        ;;PICKUP "DATA HEADER" POINTER
056364  001404                  BEQ     5$              ;;SKIP TYPEOUT IF 0
056366  104401                  TYPE                    ;;TYPE THE "DATA HEADER"
056370  000000          4$:     .WORD   0               ;;"DATA HEADER" POINTER GOES HERE
056372  104401  001223          TYPE    ,$CRLF          ;;"CARRIAGE RETURN" & "LINE FEED"
056376  010146          5$:     MOV     R1,-(SP)        ;;SAVE R1
056400  012001                  MOV     (R0)+,R1        ;;PICKUP "DATA TABLE" POINTER
056402  001415                  BEQ     9$              ;;BR IF NO DATA TO BE TYPED
056404  012000                  MOV     (R0)+,R0        ;;PICKUP "DATA FORMAT" POINTER
056406  105720          6$:     TSTB    (R0)+           ;;"OCTAL" OR "DECIMAL"
056410  001003                  BNE     7$              ;;BR IF DECIMAL
056412  013146                  MOV     @(R1)+,-(SP)    ;;SAVE @(R1)+ FOR TYPEOUT
056414  104402                  TYPOC                   ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
056416  000402                  BR      8$
056420                  7$:
056420  013146                  MOV     @(R1)+,-(SP)    ;;SAVE @(R1)+ FOR TYPEOUT
056422  104405                  TYPDS                   ;;GO TYPE--DECIMAL ASCII WITH SIGN
056424  005711          8$:     TST     (R1)            ;;IS THERE ANOTHER NUMBER?
056426  001403                  BEQ     9$              ;;BR IF NO
056430  104401  056450          TYPE    ,11$            ;;TYPE TWO(2) SPACES
056434  000764                  BR      6$              ;;LOOP

056436  012601          9$:     MOV     (SP)+,R1        ;;RESTORE R1
056440  012600          10$:    MOV     (SP)+,R0        ;;RESTORE R0
056442  104401  001223          TYPE    ,$CRLF          ;;"CARRIAGE RETURN" & "LINE FEED"
056446  000207                  RTS     PC              ;;RETURN
056450  020040    000   11$:    .ASCIZ  /  /            ;;TWO(2) SPACES
056454                          .EVEN
```

```
                                .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

056454  017646  000000         $TYPOS: MOV     @(SP),-(SP)     ;;PICKUP THE MODE
056460  116637  000001  056677         MOVB    1(SP),$OFILL    ;;LOAD ZERO FILL SWITCH
056466  112637  056701                 MOVB    (SP)+,$OMODE+1  ;;NUMBER OF DIGITS TO TYPE
056472  062716  000002                 ADD     #2,(SP)         ;;ADJUST RETURN ADDRESS
056476  000406                         BR      $TYPON
056500  112737  000001  056677 $TYPOC: MOVB    #1,$OFILL       ;;SET THE ZERO FILL SWITCH
056506  112737  000006  056701         MOVB    #6,$OMODE+1     ;;SET FOR SIX(6) DIGITS
056514  112737  000005  056676 $TYPON: MOVB    #5,$OCNT        ;;SET THE ITERATION COUNT
056522  010346                         MOV     R3,-(SP)        ;;SAVE R3
056524  010446                         MOV     R4,-(SP)        ;;SAVE R4
056526  010546                         MOV     R5,-(SP)        ;;SAVE R5
056530  113704  056701                 MOVB    $OMODE+1,R4     ;;GET THE NUMBER OF DIGITS TO TYPE
056534  005404                         NEG     R4
056536  062704  000006                 ADD     #6,R4           ;;SUBTRACT IT FOR MAX. ALLOWED
056542  110437  056700                 MOVB    R4,$OMODE       ;;SAVE IT FOR USE
056546  113704  056677                 MOVB    $OFILL,R4       ;;GET THE ZERO FILL SWITCH
056552  016605  000012                 MOV     12(SP),R5       ;;PICKUP THE INPUT NUMBER
056556  005003                         CLR     R3              ;;CLEAR THE OUTPUT WORD
056560  006105                 1$:     ROL     R5              ;;ROTATE MSB INTO "C"
056562  000404                         BR      3$              ;;GO DO MSB
056564  006105                 2$:     ROL     R5              ;;FORM THIS DIGIT
056566  006105                         ROL     R5
056570  006105                         ROL     R5
056572  010503                         MOV     R5,R3
056574  006103                 3$:     ROL     R3              ;;GET LSB OF THIS DIGIT
056576  105337  056700                 DECB    $OMODE          ;;TYPE THIS DIGIT?
056602  100016                         BPL     7$              ;;BR IF NO
056604  042703  177770                 BIC     #177770,R3      ;;GET RID OF JUNK
056610  001002                         BNE     4$              ;;TEST FOR 0
056612  005704                         TST     R4              ;;SUPPRESS THIS 0?
056614  001403                         BEQ     5$              ;;BR IF YES
```

```
11450  056616  005204                  4$:     INC     R4              ;;DON'T SUPPRESS ANYMORE 0'S
11451  056620  052703  000060                  BIS     #'0,R3          ;;MAKE THIS DIGIT ASCII
11452  056624  052703  000040          5$:     BIS     #' ,R3          ;;MAKE ASCII IF NOT ALREADY
11453  056630  110337  056674                  MOVB    R3,8$           ;;SAVE FOR TYPING
11454  056634  104401  056674                  TYPE    ,8$             ;;GO TYPE THIS DIGIT
11455  056640  105337  056676          7$:     DECB    $OCNT           ;;COUNT BY 1
11456  056644  003347                          BGT     2$              ;;BR IF MORE TO DO
11457  056646  002402                          BLT     6$              ;;BR IF DONE
11458  056650  005204                          INC     R4              ;;INSURE LAST DIGIT ISN'T A BLANK
11459  056652  000744                          BR      2$              ;;GO DO THE LAST DIGIT
11460  056654  012605          6$:     MOV     (SP)+,R5        ;;RESTORE R5
11461  056656  012604                  MOV     (SP)+,R4        ;;RESTORE R4
11462  056660  012603                  MOV     (SP)+,R3        ;;RESTORE R3
11463  056662  016666  000002  000004          MOV     2(SP),4(SP)     ;;SET THE STACK FOR RETURNING
11464  056670  012616                  MOV     (SP)+,(SP)
11465  056672  000002                  RTI                     ;;RETURN
11466  056674     000          8$:     .BYTE   0               ;;STORAGE FOR ASCII DIGIT
11467  056675     000                  .BYTE   0               ;;TERMINATOR FOR TYPE ROUTINE
11468  056676     000          $OCNT:  .BYTE   0               ;;OCTAL DIGIT COUNTER
11469  056677     000          $OFILL: .BYTE   0               ;;ZERO FILL SWITCH
11470  056700  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.

056702  010046                  $TRAP:  MOV     R0,-(SP)        ;;SAVE R0
056704  016600  000002                  MOV     2(SP),R0        ;;GET TRAP ADDRESS
056710  005740                          TST     -(R0)           ;;BACKUP BY 2
056712  111000                          MOVB    (R0),R0         ;;GET RIGHT BYTE OF TRAP
056714  006300                          ASL     R0              ;;POSITION FOR INDEXING
056716  016000  056736                  MOV     $TRPAD(R0),R0   ;;INDEX TO TABLE
056722  000200                          RTS     R0              ;;GO TO ROUTINE


                                ;;THIS IS USE TO HANDLE THE "GETPRI" MACRO

056724  011646                  $TRAP2: MOV     (SP),-(SP)      ;;MOVE THE PC DOWN
056726  016666  000004  000002          MOV     4(SP),2(SP)     ;;MOVE THE PSW DOWN
056734  000002                          RTI                     ;;RESTORE THE PSW

                                .SBTTL  TRAP TABLE

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

                                ;       ROUTINE
                                ;       -------
056736  056724                  $TRPAD: .WORD   $TRAP2
056740  054432                          $TYPE   ;;CALL=TYPE     TRAP+1(104401)  TTY TYPEOUT ROUTINE
056742  056500                          $TYPOC  ;;CALL=TYPOC    TRAP+2(104402)  TYPE OCTAL NUMBER (WITH LEADING ZEROS)
056744  056454                          $TYPOS  ;;CALL=TYPOS    TRAP+3(104403)  TYPE OCTAL NUMBER (NO LEADING ZEROS)
056746  056514                          $TYPON  ;;CALL=TYPON    TRAP+4(104404)  TYPE OCTAL NUMBER (AS PER LAST CALL)
056750  054206                          $TYPDS  ;;CALL=TYPDS    TRAP+5(104405)  TYPE DECIMAL NUMBER (WITH SIGN)

056752  055250                          $GTSWR  ;;CALL=GTSWR    TRAP+6(104406)  GET SOFT-SWR SETTING

056754  055160                          $CKSWR  ;;CALL=CKSWR    TRAP+7(104407)  TEST FOR CHANGE IN SOFT-SWR
056756  055522                          $RDCHR  ;;CALL=RDCHR    TRAP+10(104410) TTY TYPEIN CHARACTER ROUTINE
056760  055612                          $RDLIN  ;;CALL=RDLIN    TRAP+11(104411) TTY TYPEIN STRING ROUTINE
056762  055772                          $RDOCT  ;;CALL=RDOCT    TRAP+12(104412) READ AN OCTAL NUMBER FROM TTY
056764  042516                          T.SCOP  ;;CALL=SCOP1    TRAP+13(104413) MY LOCAL SCOPES
056766  042570                          CHECKT  ;;CALL=CHECKD   TRAP+14(104414) CHECK DVA,RDY,DPR,DRY
056770  043106                          WAIT.T  ;;CALL=WAT      TRAP+15(104415) WAIT LOOP
```

```
11517                                   .SBTTL  POWER DOWN AND UP ROUTINES
11518
11519                                   ;;*****************************************************************
11520                                   ;POWER DOWN ROUTINE
11521  056772  012737  057136  000024  $PWRDN: MOV     #$ILLUP,@#PWRVEC ;;SET FOR FAST UP
11522  057000  012737  000340  000026          MOV     #340,@#PWRVEC+2 ;;PRIO:7
11523  057006  010046                         MOV     R0,-(SP)        ;;PUSH R0 ON STACK
11524  057010  010146                         MOV     R1,-(SP)        ;;PUSH R1 ON STACK
11525  057012  010246                         MOV     R2,-(SP)        ;;PUSH R2 ON STACK
11526  057014  010346                         MOV     R3,-(SP)        ;;PUSH R3 ON STACK
11527  057016  010446                         MOV     R4,-(SP)        ;;PUSH R4 ON STACK
11528  057020  010546                         MOV     R5,-(SP)        ;;PUSH R5 ON STACK
11529  057022  017746  122112                 MOV     @SWR,-(SP)      ;;PUSH @SWR ON STACK
11530  057026  010637  057142                 MOV     SP,$SAVR6       ;;SAVE SP
11531  057032  012737  057044  000024         MOV     #$PWRUP,@#PWRVEC ;;SET UP VECTOR
11532  057040  000000                         HALT
11533  057042  000776                         BR      .-2             ;;HANG UP
11534
11535                                   ;;*****************************************************************
11536                                   ;POWER UP ROUTINE
11537  057044  012737  057136  000024  $PWRUP: MOV     #$ILLUP,@#PWRVEC ;;SET FOR FAST DOWN
11538  057052  013706  057142                 MOV     $SAVR6,SP       ;;GET SP
11539  057056  005037  057142                 CLR     $SAVR6          ;;WAIT LOOP FOR THE TTY
11540  057062  005237  057142          1$:    INC     $SAVR6          ;;WAIT FOR THE INC
11541  057066  001375                         BNE     1$              ;;OF  WORD
11542  057070  012677  122044                 MOV     (SP)+,@SWR      ;;POP STACK INTO @SWR
11543  057074  012605                         MOV     (SP)+,R5        ;;POP STACK INTO R5
11544  057076  012604                         MOV     (SP)+,R4        ;;POP STACK INTO R4
11545  057100  012603                         MOV     (SP)+,R3        ;;POP STACK INTO R3
11546  057102  012602                         MOV     (SP)+,R2        ;;POP STACK INTO R2
11547  057104  012601                         MOV     (SP)+,R1        ;;POP STACK INTO R1
11548  057106  012600                         MOV     (SP)+,R0        ;;POP STACK INTO R0
11549  057110  012737  056772  000024         MOV     #$PWRDN,@#PWRVEC ;;SET UP THE POWER DOWN VECTOR
11550  057116  012737  000340  000026         MOV     #340,@#PWRVEC+2 ;;PRIO:7
11551  057124  104401                         TYPE                    ;REPORT THE POWER FAILURE
11552  057126  057144                 $PWRMG: .WORD   $POWER           ;;POWER FAIL MESSAGE POINTER
11553  057130  012716                         MOV     (PC)+,(SP)      ;;RESTART AT BEGIN
11554  057132  004240                 $PWRAD: .WORD   BEGIN           ;;RESTART ADDRESS
11555  057134  000002                         RTI
11556  057136  000000                 $ILLUP: HALT                    ;;THE POWER UP SEQUENCE WAS STARTED
11557  057140  000776                         BR      .-2             ;; BEFORE THE POWER DOWN WAS COMPLETE
11558  057142  000000                 $SAVR6: 0                       ;;PUT THE SP HERE
11559  057144  005015  047520  042527 $POWER: .ASCIZ  <15><12>"POWER"
11560  057152  000122
11561                                           .EVEN
```

BL

```
11562                                ;*****************************************************************************
11563                                ;*
11564                                ;*ERROR AND MESSAGE TABLE CONDIMENTS
11565                                ;*
11566                                ;*****************************************************************************
11567
11568
11569
11570
11571  057154  051127  047117  020107  EM1:    .ASCIZ  /WRONG DATA IN READING OR WRITING HARDWARE REGISTER/
11572  057162  040504  040524  044440
11573  057170  020116  042522  042101
11574  057176  047111  020107  051117
11575  057204  053440  044522  044524
11576  057212  043516  044040  051101
11577  057220  053504  051101  020105
11578  057226  042522  044507  052123
11579  057234  051105     000
11580  057237     105  051122  051117  EM2:    .ASCIZ  /ERROR ON  DATA COMMAND/
11581  057244  047440  020116  042040
11582  057252  052101  020101  047503
11583  057260  046515  047101  000104
11584  057266  051105  047522  020122  EM6:    .ASCIZ  /ERROR ON WRITE HEADER AND DATA/
11585  057274  047117  053440  044522
11586  057302  042524  044040  040505
11587  057310  042504  020122  047101
11588  057316  020104  040504  040524
11589  057324     000
11590  057325     103  047117  051124  EM11:   .ASCIZ  /CONTROLLER OR DRIVE STATUS/
11591  057332  046117  042514  020122
11592  057340  051117  042040  044522
11593  057346  042526  051440  040524
11594  057354  052524  000123
11595  057360  042522  044507  052123  EM14:   .ASCIZ  /REGISTER FAILED/
11596  057366  051105  043040  044501
11597  057374  042514  000104
11598  057400  050123  041505  043111  EM15:   .ASCIZ  /SPECIFIED REGISTER NON EXISTANT SO ABORT/
11599  057406  042511  020104  042522
11600  057414  044507  052123  051105
11601  057422  047040  047117  042440
11602  057430  044530  052123  047101
11603  057436  020124  047523  040440
11604  057444  047502  052122     000
11605  057451     127  044501  020124  EM16:   .ASCIZ  /WAIT LOOP FAILED/
11606  057456  047514  050117  043040
11607  057464  044501  042514  000104
11608  057472  051127  052111  020105  EM17:   .ASCIZ  /WRITE CHECK FAILING/
11609  057500  044103  041505  020113
11610  057506  040506  046111  047111
11611  057514  000107
11612  057516  042522  044507  052123  EM20:   .ASCIZ  /REGISTER FAILING/
11613  057524  051105  043040  044501
11614  057532  044514  043516     000
11615  057537     111  052116  051105  EM21:   .ASCIZ  /INTERRUPT FAILING/
11616  057544  052522  052120  043040
11617  057552  044501  044514  043516
```

C4

```
11618  057560     000
11619  057561     105  051122  051117  EM22:   .ASCII  /ERROR ON DRIVES PRESENT -/<15><12>
11620  057566  047440  020116  051104
11621  057574  053111  051505  050040
11622  057602  042522  042523  052116
11623  057610  026440  005015
11624  057614  044124  020105  047125          .ASCII  /THE UNIT NO'S FOUND BY SETTING RHAS USING RHER1/<15><12>
11625  057622  052111  047040  023517
11626  057630  020123  047506  047125
11627  057636  020104  054502  051440
11628  057644  052105  044524  043516
11629  057652  051040  040510  020123
11630  057660  051525  047111  020107
11631  057666  044122  051105  006461
11632  057674     012
11633  057675     050  032124  020051          .ASCII  /(T4) DO NOT AGREE WITH THE UNIT NO'S FOUND/<15><12>
11634  057702  047504  047040  052117
11635  057710  040440  051107  042505
11636  057716  053440  052111  020110
11637  057724  044124  020105  047125
11638  057732  052111  047040  023517
11639  057740  020123  047506  047125
11640  057746  006504     012
11641  057751     102  020131  047514          .ASCII  /BY LOOKING FOR 'NED' = 0 IN RHCS2 (BIT #12)/<15><12><15><12>
11642  057756  045517  047111  020107
11643  057764  047506  020122  047047
11644  057772  042105  020047  020075
11645  060000  020060  047111  051040
11646  060006  041510  031123  024040
11647  060014  044502  020124  030443
11648  060022  024462  005015  005015
11649  060030  047516  042524  020072          .ASCII  /NOTE: ON DUAL PORT SYSTEM, A DRIVE ON OTHER PORT WILL /<15><12>
11650  060036  047117  042040  040525
11651  060044  020114  047520  052122
11652  060052  051440  051531  042524
11653  060060  026115  040440  042040
11654  060066  044522  042526  047440
11655  060074  020116  052117  042510
11656  060102  020122  047520  052122
11657  060110  053440  046111  020114
11658  060116  005015
11659  060120  047516  020124  044507          .ASCII  /NOT GIVE 'NED', BUT WILL GIVE RHAS RESPONSES/<15><12>
11660  060126  042526  023440  042516
11661  060134  023504  020054  052502
11662  060142  020124  044527  046114
11663  060150  043440  053111  020105
11664  060156  044122  051501  051040
11665  060164  051505  047520  051516
11666  060172  051505  005015
11667  060176  042510  041516  020105          .ASCIZ  /HENCE THERE WILL BE AN EXTRA DRIVE/
11668  060204  044124  051105  020105
11669  060212  044527  046114  041040
11670  060220  020105  047101  042440
11671  060226  052130  040522  042040
11672  060234  044522  042526     000
11673  060241     114  047517  020113  EM24:   .ASCIZ  /LOOK AHEAD REGISTER AT THE BEGINNING OF SECTOR IS IN ERROR/
```

D6

```
11674  060246  044101  040505  020104
11675  060254  042522  044507  052123
11676  060262  051105  040440  020124
11677  060270  044124  020105  042502
11678  060276  044507  047116  047111
11679  060304  020107  043117  051440
11680  060312  041505  047524  020122
11681  060320  051511  044440  020116
11682  060326  051105  047522  000122
11683  060334  047514  045517  040440  EM25:   .ASCIZ  /LOOK AHEAD REGISTER IS IN ERROR/
11684  060342  042510  042101  051040
11685  060350  043505  051511  042524
11686  060356  020122  051511  044440
11687  060364  020116  051105  047522
11688  060372  000122
11689  060374  052503  051122  047105  EM30:   .ASCII  /CURRENT CYLINDER DOES NOT MATCH DESIRED CYLINDER REGIHSTER/<15><12>
11690  060402  020124  054503  044514
11691  060410  042116  051105  042040
11692  060416  042517  020123  047516
11693  060424  020124  040515  041524
11694  060432  020110  042504  044523
11695  060440  042522  020104  054503
11696  060446  044514  042116  051105
11697  060454  051040  043505  044111
11698  060462  052123  051105  005015
11699  060470  043101  042524  020122          .ASCIZ  /AFTER A SEEK AND INIT/
11700  060476  020101  042523  045505
11701  060504  040440  042116  044440
11702  060512  044516  000124
11703  060516  041505  020103  042507  EM31:   .ASCII  /ECC GENERATED IS INCORRECT/<15><12>
11704  060524  042516  040522  042524
11705  060532  020104  051511  044440
11706  060540  041516  051117  042522
11707  060546  052103  005015
11708  060552  053105  051105  020131          .ASCIZ  /EVERY WORD ON THIS SECTOR IS THAT GIVEN IN "DATA USED"/
11709  060560  047527  042122  047440
11710  060566  020116  044124  051511
11711  060574  051440  041505  047524
11712  060602  020122  051511  052040
11713  060610  040510  020124  044507
11714  060616  042526  020116  047111
11715  060624  021040  040504  040524
11716  060632  052440  042523  021104
11717  060640     000
11718  060641     117  020116  042522  EM32:   .ASCII  /ON READ COMMAND, AFTER DATA AND ECC HAVE BEEN READ,/<15><12>
11719  060646  042101  041440  046517
11720  060654  040515  042116  020054
11721  060662  043101  042524  020122
11722  060670  040504  040524  040440
11723  060676  042116  042440  041503
11724  060704  044040  053101  020105
11725  060712  042502  047105  051040
11726  060720  040505  026104  005015
11727  060726  041505  020103  042522          .ASCII  /ECC REGISTERS OR RHER1 ARE IN ERROR/<15><12>
11728  060734  044507  052123  051105
11729  060742  020123  051117  051040
```

```
11730  060750  042510  030522  040440
11731  060756  042522  044440  020116
11732  060764  051105  047522  006522
11733  060772     012
11734  060773     117  046116  020131          .ASCII  /ONLY LOWER 11 BITS OF PATTERN REG. CAN BE READ/<15><12>
11735  061000  047514  042527  020122
11736  061006  030461  041040  052111
11737  061014  020123  043117  050040
11738  061022  052101  042524  047122
11739  061030  051040  043505  020056
11740  061036  040503  020116  042502
11741  061044  051040  040505  006504
11742  061052     012
11743  061053     124  044510  020123          .ASCIZ  /THIS SHOULD MATCH LOWER 11 BITS OF GOOD ECC1/
11744  061060  044123  052517  042114
11745  061066  046440  052101  044103
11746  061074  046040  053517  051105
11747  061102  030440  020061  044502
11748  061110  051524  047440  020106
11749  061116  047507  042117  042440
11750  061124  041503  000061
11751  061130  044510  044107  041440  EM33:   .ASCIZ  /HIGH COUNT BIT NOT SET AFTER 38659 CLOCKS/
11752  061136  052517  052116  041040
11753  061144  052111  047040  052117
11754  061152  051440  052105  040440
11755  061160  052106  051105  031440
11756  061166  034070  034465  041440
11757  061174  047514  045503  000123
11758  061202  042532  047522  042040  EM34:   .ASCIZ  /ZERO DETECT BIT NOT HIGH WHEN 32 BIT ECC REG. HAS 21 ZEROS/
11759  061210  052105  041505  020124
11760  061216  044502  020124  047516
11761  061224  020124  044510  044107
11762  061232  053440  042510  020116
11763  061240  031063  041040  052111
11764  061246  042440  041503  051040
11765  061254  043505  020056  040510
11766  061262  020123  030462  055040
11767  061270  051105  051517     000
11768  061275     120  051517  052111  EM35:   .ASCII  /POSITION REGISTER OR 11 BITS OF PATTERN REGISTER INCORRECT/<15><12>
11769  061302  047511  020116  042522
11770  061310  044507  052123  051105
11771  061316  047440  020122  030461
11772  061324  041040  052111  020123
11773  061332  043117  050040  052101
11774  061340  042524  047122  051040
11775  061346  043505  051511  042524
11776  061354  020122  047111  047503
11777  061362  051122  041505  006524
11778  061370     012
11779  061371     114  053517  051105          .ASCII  /LOWER 11 BITS OF PATTERN REGISTER SHOULD MATCH LOWER/<15><12>
11780  061376  030440  020061  044502
11781  061404  051524  047440  020106
11782  061412  040520  052124  051105
11783  061420  020116  042522  044507
11784  061426  052123  051105  051440
11785  061434  047510  046125  020104
```

```
11786  061442  040515  041524  020110
11787  061450  047514  042527  006522
11788  061456     012
11789  061457     061  020061  044502          .ASCII  /11 BITS OF GOOD ECC1/<15><12>
11790  061464  051524  047440  020106
11791  061472  047507  042117  042440
11792  061500  041503  006461     012
11793  061505     104  052101  042440          .ASCIZ  /DAT ENVLOP GOOD POSITION AND N-CODE ZEROS ARE IN OCTAL/
11794  061512  053116  047514  020120
11795  061520  047507  042117  050040
11796  061526  051517  052111  047511
11797  061534  020116  047101  020104
11798  061542  026516  047503  042504
11799  061550  055040  051105  051517
11800  061556  040440  042522  044440
11801  061564  020116  041517  040524
11802  061572  000114
11803  061574  047117  051040  040505  EM36:   .ASCIZ  /ON READ COMMAND WITH NON-CORRECTABLE ERROR "DCK" AND "ECH" SHOULD BE SE
11804  061602  020104  047503  046515
11805  061610  047101  020104  044527
11806  061616  044124  047040  047117
11807  061624  041455  051117  042522
11808  061632  052103  041101  042514
11809  061640  042440  051122  051117
11810  061646  023440  041504  023513
11811  061654  040440  042116  023440
11812  061662  041505  023510  051440
11813  061670  047510  046125  020104
11814  061676  042502  051440  052105
11815  061704     000
11816  061705     120  047522  051107  EM37:   .ASCII  /PROGRAM ERROR BIT #10 IN RHCS2 DID NOT SET/<15><12>
11817  061712  046501  042440  051122
11818  061720  051117  041040  052111
11819  061726  021440  030061  044440
11820  061734  020116  044122  051503
11821  061742  020062  044504  020104
11822  061750  047516  020124  042523
11823  061756  006524     012
11824  061761     111  020106  047520          .ASCIZ  /IF POSITION REGISTER =10040 OR 10041, IT IS GOOD/
11825  061766  044523  044524  047117
11826  061774  051040  043505  051511
11827  062002  042524  020122  030475
11828  062010  030060  030064  047440
11829  062016  020122  030061  032060
11830  062024  026061  044440  020124
11831  062032  051511  043440  047517
11832  062040  000104
11833
11834  062042  044122  041527  042040  EM40:   .ASCII  /RHWC DID NOT = 0 UPON COMPLETION OF READ/<15><12>
11835  062050  042111  047040  052117
11836  062056  036440  030040  052440
11837  062064  047520  020116  047503
11838  062072  050115  042514  044524
11839  062100  047117  047440  020106
11840  062106  042522  042101  005015
11841  062114  051117  053440  044522          .ASCIZ  /OR WRITE HEADER AND DATA/
```

```
 11842  062122  042524  044040  040505
 11843  062130  042504  020122  047101
 11844  062136  020104  040504  040524
 11845  062144     000
```

```
11846
11847  062145    106  052101  046101  CPHALT: .ASCII  /FATAL ERROR - SEE DOCUMENT LISTING/<15><12>
11848  062152 042440  051122  051117
11849  062160 026440  051440  042505
11850  062166 042040  041517  046525
11851  062174 047105  020124  044514
11852  062202 052123  047111  006507
11853  062210    012
11854  062211    040  005015  177607          .ASCII  / /<15><12><207><377><377><207><377><377><207><377><377>
11855  062216 103777  177777  177607
11856  062224    377
11857  062225    124  042510  041440          .ASCII  /THE CONTROLLER OR DEVICE HAS GONE OFFLINE, LOST/<15><12>
11858  062232 047117  051124  046117
11859  062240 042514  020122  051117
11860  062246 042040  053105  041511
11861  062254 020105  040510  020123
11862  062262 047507  042516  047440
11863  062270 043106  044514  042516
11864  062276 020054  047514  052123
11865  062304 005015
11866  062306 051047  040505  054504          .ASCII  /'READY', BECOME UNAVAILABLE, OR HAS STATUS BITS/<15><12>
11867  062314 026047  041040  041505
11868  062322 046517  020105  047125
11869  062330 053101  044501  040514
11870  062336 046102  026105  047440
11871  062344 020122  040510  020123
11872  062352 052123  052101  051525
11873  062360 041040  052111  006523
11874  062366    012
11875  062367    127  044510  044103          .ASCIZ  /WHICH CANNOT BE CLEARED/
11876  062374 041440  047101  047516
11877  062402 020124  042502  041440
11878  062410 042514  051101  042105
11879  062416    000
11880
11881  062417    040  020040  020040  SPACE1: .ASCII  /      /
11882  062424    040
11883  062425    040  000040          SPACE2: .ASCIZ  / /
11884
11885
11886  062430 041520  020040  020040  DH1:    .ASCII  /PC      TEST    REG.    GOOD    RECEIVED/<15><12>
11887  062436 020040  042524  052123
11888  062444 020040  020040  042522
11889  062452 027107  020040  020040
11890  062460 047507  042117  020040
11891  062466 020040  042522  042503
11892  062474 053111  042105  005015
11893  062502 020040  020040  020040          .ASCIZ  /        NO      ADDR.   DATA    DATA    /
11894  062510 020040  047516  020040
11895  062516 020040  020040  042101
11896  062524 051104  020056  020040
11897  062532 040504  040524  020040
11898  062540 020040  040504  040524
11899  062546 020040  020040     000
11900  062553    120  020103  020040  DH2:    .ASCII  /PC      TEST    WORD    GOOD    BAD     /<15><12>
11901  062560 020040  052040  051505
```

```
11902  062566  020124  020040  053440
11903  062574  051117  020104  020040
11904  062602  043440  047517  020104
11905  062610  020040  041040  042101
11906  062616  020040  020040  006440
11907  062624     012
11908  062625     040  020040  020040          .ASCIZ  /       NO      NO      DATA    DATA    /
11909  062632  020040  047040  020117
11910  062640  020040  020040  047040
11911  062646  020117  020040  020040
11912  062654  042040  052101  020101
11913  062662  020040  042040  052101
11914  062670  020101  020040  000040
11915  062676  041520  020040  020040  DH11:   .ASCII  /PC     TEST    FAILING CONT.   CONT.   CONT.   CONT.   /<15><12>
11916  062704  020040  042524  052123
11917  062712  020040  020040  040506
11918  062720  046111  047111  020107
11919  062726  047503  052116  020056
11920  062734  020040  047503  052116
11921  062742  020056  020040  047503
11922  062750  052116  020056  020040
11923  062756  047503  052116  020056
11924  062764  020040  005015
11925  062770  020040  020040  020040          .ASCIZ  /       NO      REG ADR RHCS1   RHCS2   RHDS1   RHER1/
11926  062776  020040  047516  020040
11927  063004  020040  020040  042522
11928  063012  020107  042101  020122
11929  063020  044122  051503  020061
11930  063026  020040  044122  051503
11931  063034  020062  020040  044122
11932  063042  051504  020061  020040
11933  063050  044122  051105  000061
11934  063056  041520  020040  020040  DH14:   .ASCII  /PC     TEST    FAILING CONT.   CONT.   CONT.   CONT.   CONT.   /<15><1
11935  063064  020040  042524  052123
11936  063072  020040  020040  040506
11937  063100  046111  047111  020107
11938  063106  047503  052116  020056
11939  063114  020040  047503  052116
11940  063122  020056  020040  047503
11941  063130  052116  020056  020040
11942  063136  047503  052116  020056
11943  063144  020040  047503  052116
11944  063152  020056  020040  005015
11945  063160  020040  020040  020040          .ASCIZ  /       NO      REG ADR BAD REG RHCS1   RHCS2   RHDS1   RHER1/
11946  063166  020040  047516  020040
11947  063174  020040  020040  042522
11948  063202  020107  042101  020122
11949  063210  040502  020104  042527
11950  063216  020107  044122  051503
11951  063224  020061  020040  044122
11952  063232  051503  020062  020040
11953  063240  044122  051504  020061
11954  063246  020040  044122  051105
11955  063254  000061
11956  063256  041520  020040  020040  DH15:   .ASCIZ  /PC     TEST    REG ADR /
11957  063264  020040  042524  052123
```

```
11958  063272  020040  020040  042522
11959  063300  020107  042101  020122
11960  063306     000
11961  063307     120  020103  020040  DH16:  .ASCII  /PC      TEST    WAIT    BIT     REG     REG     /<15><12>
11962  063314  020040  052040  051505
11963  063322  020124  020040  053440
11964  063330  044501  020124  020040
11965  063336  041040  052111  020040
11966  063344  020040  051040  043505
11967  063352  020040  020040  051040
11968  063360  043505  020040  020040
11969  063366  006440     012
11970  063371     040  020040  020040         .ASCIZ  /       NO      PC      WANTED  ADDRESS CONT.   /
11971  063376  020040  047040  020117
11972  063404  020040  020040  050040
11973  063412  020103  020040  020040
11974  063420  053440  047101  042524
11975  063426  020104  040440  042104
11976  063434  042522  051523  041440
11977  063442  047117  027124  020040
11978  063450  000040
11979  063452  041520  020040  020040  DH17:  .ASCII  /PC      TEST    CONT.   CONT.   CONT.   CONT.   CONT.   /<15><12>
11980  063460  020040  042524  052123
11981  063466  020040  020040  047503
11982  063474  052116  020056  020040
11983  063502  047503  052116  020056
11984  063510  020040  047503  052116
11985  063516  020056  020040  047503
11986  063524  052116  020056  020040
11987  063532  047503  052116  020056
11988  063540  020040  005015
11989  063544  020040  020040  020040         .ASCIZ  /       NO      RHBA    RHDB    RHWC    RHCS1   RHCS2   /
11990  063552  020040  047516  020040
11991  063560  020040  020040  044122
11992  063566  040502  020040  020040
11993  063574  044122  041104  020040
11994  063602  020040  044122  041527
11995  063610  020040  020040  044122
11996  063616  051503  020061  020040
11997  063624  044122  051503  020062
11998  063632  020040     000
11999  063635     120  020103  020040  DH20:  .ASCII  /PC      TEST    CONT    CONT    CONT    CONT    CONT    /<15><12>
12000  063642  020040  052040  051505
12001  063650  020124  020040  041440
12002  063656  047117  020124  020040
12003  063664  041440  047117  020124
12004  063672  020040  041440  047117
12005  063700  020124  020040  041440
12006  063706  047117  020124  020040
12007  063714  041440  047117  020124
12008  063722  020040  006440     012
12009  063727     040  020040  020040         .ASCIZ  /       NO      PHER1   PHER2   PHER3   RHAS    RHDS1   /
12010  063734  020040  047040  020117
12011  063742  020040  020040  051040
12012  063750  042510  030522  020040
12013  063756  051040  042510  031122
```

KL

```
12014  063764  020040  051040  042510
12015  063772  031522  020040  051040
12016  064000  040510  020123  020040
12017  064006  051040  042110  030523
12018  064014  020040  000040
12019  064020  041520  020040  020040  DH21:   .ASCII  /PC      TEST    CONT    CONT    CONT    /<15><12>
12020  064026  020040  042524  052123
12021  064034  020040  020040  047503
12022  064042  052116  020040  020040
12023  064050  047503  052116  020040
12024  064056  020040  047503  052116
12025  064064  020040  020040  005015
12026  064072  020040  020040  020040          .ASCIZ  /       NO      RHCS1   RHAS    RHDS1/
12027  064100  020040  047516  020040
12028  064106  020040  020040  044122
12029  064114  051503  020061  020040
12030  064122  044122  051501  020040
12031  064130  020040  044122  051504
12032  064136  000061
12033  064140  041520  020040  020040  DH22:   .ASCII  /PC      TEST       RHAS    RHCS2/<15><12>
12034  064146  020040  042524  052123
12035  064154  020040  020040  020040
12036  064162  051040  040510  020123
12037  064170  020040  051040  041510
12038  064176  031123  005015
12039  064202  020040  020040  020040          .ASCIZ  /       NO         UNITS   UNITS/
12040  064210  020040  047516  020040
12041  064216  020040  020040  020040
12042  064224  052440  044516  051524
12043  064232  020040  052440  044516
12044  064240  051524     000
12045  064243     120  020103  020040  DH24:   .ASCII  /PC      TEST    RHDST   BAD     GOOD    SECTOR  SECTOR  /<15><12>
12046  064250  020040  052040  051505
12047  064256  020124  020040  051040
12048  064264  042110  052123  020040
12049  064272  041040  042101  020040
12050  064300  020040  043440  047517
12051  064306  020104  020040  051440
12052  064314  041505  047524  020122
12053  064322  051440  041505  047524
12054  064330  020122  006440     012
12055  064335     040  020040  020040          .ASCIZ  /       NO      CONT.   RHLA    RHLA    NO      CLOCK   /
12056  064342  020040  047040  020117
12057  064350  020040  020040  041440
12058  064356  047117  027124  020040
12059  064364  051040  046110  020101
12060  064372  020040  051040  046110
12061  064400  020101  020040  047040
12062  064406  020117  020040  020040
12063  064414  041440  047514  045503
12064  064422  020040  000040
12065  064426  041520  020040  020040  DH26:   .ASCII  /PC      TEST    PC OF   FAILING CONT.   CONT.   CONT.   CONT.   /<15><1
12066  064434  020040  042524  052123
12067  064442  020040  020040  041520
12068  064450  047440  020106  020040
12069  064456  040506  046111  047111
```

```
12070  064464  020107  047503  052116
12071  064472  020056  020040  047503
12072  064500  052116  020056  020040
12073  064506  047503  052116  020056
12074  064514  020040  047503  052116
12075  064522  020056  020040  005015
12076  064530  020040  020040  020040          .ASCIZ  /       NO      JSR     REG ADD RHCS1   RHCS2   RHDS1   RHER1   /
12077  064536  020040  047516  020040
12078  064544  020040  020040  051512
12079  064552  020122  020040  020040
12080  064560  042522  020107  042101
12081  064566  020104  044122  051503
12082  064574  020061  020040  044122
12083  064602  051503  020062  020040
12084  064610  044122  051504  020061
12085  064616  020040  044122  051105
12086  064624  020061  020040     000
12087  064631     120  020103  020040  DH27:   .ASCII  /PC      TEST    PC OF   FAILING GOOD    RECEIVED/<15><12>
12088  064636  020040  052040  051505
12089  064644  020124  020040  050040
12090  064652  020103  043117  020040
12091  064660  043040  044501  044514
12092  064666  043516  043440  047517
12093  064674  020104  020040  051040
12094  064702  041505  044505  042520
12095  064710  006504     012
12096  064713     040  020040  020040          .ASCIZ  /       NO      JSR     REG     DATA    DATA    /
12097  064720  020040  047040  020117
12098  064726  020040  020040  045040
12099  064734  051123  020040  020040
12100  064742  051040  043505  020040
12101  064750  020040  042040  052101
12102  064756  020101  020040  042040
12103  064764  052101  020101  020040
12104  064772  000040
12105  064774  041520  020040  020040  DH30:   .ASCII  /PC      TEST    PC OF   REG.    GOOD    BAD     /<15><12>
12106  065002  020040  042524  052123
12107  065010  020040  020040  041520
12108  065016  047440  020106  020040
12109  065024  042522  027107  020040
12110  065032  020040  047507  042117
12111  065040  020040  020040  040502
12112  065046  020104  020040  020040
12113  065054  005015
12114  065056  020040  020040  020040          .ASCIZ  /       NO      JSR     ADDRESS DATA    DATA    /
12115  065064  020040  047516  020040
12116  065072  020040  020040  051512
12117  065100  020122  020040  020040
12118  065106  042101  051104  051505
12119  065114  020123  040504  040524
12120  065122  020040  020040  040504
12121  065130  040524  020040  020040
12122  065136     000
12123  065137     120  020103  020040  DH31:   .ASCII  /PC      TEST    WORD    GOOD    BAD     CONT.   CONT.   CONT    /<15><1
12124  065144  020040  052040  051505
12125  065152  020124  020040  053440
```

mb

```
12126  065160  051117  020104  020040
12127  065166  043440  047517  020104
12128  065174  020040  041040  042101
12129  065202  020040  020040  041440
12130  065210  047117  027124  020040
12131  065216  041440  047117  027124
12132  065224  020040  041440  047117
12133  065232  020124  020040  006440
12134  065240     012
12135  065241     040  020040  020040          .ASCIZ  /       NO      NO      DATA    DATA    RHCS1   RHDS1   RHER1   /
12136  065246  020040  047040  020117
12137  065254  020040  020040  047040
12138  065262  020117  020040  020040
12139  065270  042040  052101  020101
12140  065276  020040  042040  052101
12141  065304  020101  020040  051040
12142  065312  041510  030523  020040
12143  065320  051040  042110  030523
12144  065326  020040  051040  042510
12145  065334  030522  020040  000040
12146  065342  041520  020040  020040  DH32:   .ASCII  /PC      TEST    PC OF   WORD    GOOD    BAD     CONT.   CONT.   CONT.
12147  065350  020040  042524  052123
12148  065356  020040  020040  041520
12149  065364  047440  020106  020040
12150  065372  047527  042122  020040
12151  065400  020040  047507  042117
12152  065406  020040  020040  040502
12153  065414  020104  020040  020040
12154  065422  047503  052116  020056
12155  065430  020040  047503  052116
12156  065436  020056  020040  047503
12157  065444  052116  020056  020040
12158  065452  005015
12159  065454  020040  020040  020040          .ASCIZ  /       NO      JSR     NO      DATA    DATA    RHCS1   RHDS1   RHER1
12160  065462  020040  047516  020040
12161  065470  020040  020040  051512
12162  065476  020122  020040  020040
12163  065504  047516  020040  020040
12164  065512  020040  040504  040524
12165  065520  020040  020040  040504
12166  065526  040524  020040  020040
12167  065534  044122  051503  020061
12168  065542  020040  044122  051504
12169  065550  020061  020040  044122
12170  065556  051105  020061  020040
12171  065564     000
12172  065565     120  020103  020040  DH33:   .ASCII  /PC      TEST    PC OF   WORD    GOOD    CONT.   CONT.   CONT.   /<15><1
12173  065572  020040  052040  051505
12174  065600  020124  020040  050040
12175  065606  020103  043117  020040
12176  065614  053440  051117  020104
12177  065622  020040  043440  047517
12178  065630  020104  020040  041440
12179  065636  047117  027124  020040
12180  065644  041440  047117  027124
12181  065652  020040  041440  047117
```

```
12182  065660  027124  020040  006440
12183  065666     012
12184  065667     040  020040  020040          .ASCIZ  /       NO      JSR     NO      DATA    RHCS1   RHDS1   RHER1   /
12185  065674  020040  047040  020117
12186  065702  020040  020040  045040
12187  065710  051123  020040  020040
12188  065716  047040  020117  020040
12189  065724  020040  042040  052101
12190  065732  020101  020040  051040
12191  065740  041510  030523  020040
12192  065746  051040  042110  030523
12193  065754  020040  051040  042510
12194  065762  030522  020040  000040
12195  065770  041520  020040  020040  DH34:   .ASCII  /PC      TEST    GOOD    GOOD    WRITTEN WRITTEN DATA    /<15><12>
12196  065776  020040  042524  052123
12197  066004  020040  020040  047507
12198  066012  042117  020040  020040
12199  066020  047507  042117  020040
12200  066026  020040  051127  052111
12201  066034  042524  020116  051127
12202  066042  052111  042524  020116
12203  066050  040504  040524  020040
12204  066056  020040  005015
12205  066062  020040  020040  020040          .ASCIZ  /       NO      ECC1    ECC2    ECC1    ECC2    USED    /
12206  066070  020040  047516  020040
12207  066076  020040  020040  041505
12208  066104  030503  020040  020040
12209  066112  041505  031103  020040
12210  066120  020040  041505  030503
12211  066126  020040  020040  041505
12212  066134  031103  020040  020040
12213  066142  051525  042105  020040
12214  066150  020040     000
12215  066153     120  020103  020040  DH35:   .ASCII  /PC      TEST    GOOD    GOOD    PATTERN POSITON GOOD    RHER1   /<15><1
12216  066160  020040  052040  051505
12217  066166  020124  020040  043440
12218  066174  047517  020104  020040
12219  066202  043440  047517  020104
12220  066210  020040  050040  052101
12221  066216  042524  047122  050040
12222  066224  051517  052111  047117
12223  066232  043440  047517  020104
12224  066240  020040  051040  042510
12225  066246  030522  020040  006440
12226  066254     012
12227  066255     040  020040  020040          .ASCIZ  /       NO      ECC1    ECC2    REG.REG.        POSITON REG.    /
12228  066262  020040  047040  020117
12229  066270  020040  020040  042440
12230  066276  041503  020061  020040
12231  066304  042440  041503  020062
12232  066312  020040  051040  043505
12233  066320  051056  043505  020056
12234  066326  020040  020040  020040
12235  066334  050040  051517  052111
12236  066342  047117  051040  043505
12237  066350  020056  020040  000040
```

```
12238  066356  041520  020040  020040  DH36:   .ASCII  /PC      TEST   PC OF   RHWR    POSITON PATTERN /<15><12>
12239  066364  020040  042524  052123
12240  066372  020040  020040  041520
12241  066400  047440  020106  020040
12242  066406  044122  051115  020040
12243  066414  020040  047520  044523
12244  066422  047524  020116  040520
12245  066430  052124  051105  020116
12246  066436  005015
12247  066440  020040  020040  020040          .ASCIZ  /       NO     JSR     CONT.   REG.    REG./
12248  066446  020040  047516  020040
12249  066454  020040  020040  051512
12250  066462  020122  020040  020040
12251  066470  047503  052116  020056
12252  066476  020040  042522  027107
12253  066504  020040  020040  042522
12254  066512  027107     000
12255  066515     120  020103  020040  DH37:   .ASCII  /PC      TEST   POSITON POSITON GOOD    GOOD    PATTERN DATA    M-CODE/
12256  066522  020040  052040  051505
12257  066530  020124  020040  050040
12258  066536  051517  052111  047117
12259  066544  050040  051517  052111
12260  066552  047117  043440  047517
12261  066560  020104  020040  043440
12262  066566  047517  020104  020040
12263  066574  050040  052101  042524
12264  066602  047122  042040  052101
12265  066610  020101  020040  047040
12266  066616  041455  042117  006505
12267  066624     012
12268  066625     040  020040  020040          .ASCIZ  /       NO     ECC     GOOD    ECC1    ECC2    ECC     ENVLOPE ZEROS/
12269  066632  020040  047040  020117
12270  066640  020040  020040  042440
12271  066646  041503  020040  020040
12272  066654  043440  047517  020104
12273  066662  020040  042440  041503
12274  066670  020061  020040  042440
12275  066676  041503  020062  020040
12276  066704  042440  041503  020040
12277  066712  020040  042440  053116
12278  066720  047514  042520  055040
12279  066726  051105  051517     000
12280
12281  066733     120  020103  020040  DH40:   .ASCII  /PC      TEST   CONT./<15><12>
12282  066740  020040  052040  051505
12283  066746  020124  020040  041440
12284  066754  047117  027124  005015
12285  066762  020040  020040  020040          .ASCIZ  /       NO     RHWC/
12286  066770  020040  047516  020040
12287  066776  020040  020040  044122
12288  067004  041527     000
12289
12290          067010                  .EVEN
12291
12292  067010  001116  002032  042204  DT1:    .WORD   $ERRPC,TSTNM,REGADR,$GDDAT,$BDDAT,0
12293  067016  001124  001126  000000
```

```
12294  067024  001116  002032  047320  DT2:    .WORD   $ERRPC,TSTNM,ERWORD,$GDDAT,0
12295  067032  001124  000000
12296  067036  001116  002032  047320  DT3:    .WORD   $ERRPC,TSTNM,ERWORD,$GDDAT,$BDDAT,0
12297  067044  001124  001126  000000
12298
12299  067052  001116  002032  001122  DT11:   .WORD   $ERRPC,TSTNM,$BDADR,CS1,CS2,DS1,ER1,0
12300  067060  001714  001712  001736
12301  067066  001716  000000
12302  067072  001116  002032  001122  DT14:   .WORD   $ERRPC,TSTNM,$BDADR,$BDDAT,CS1,CS2,DS1,ER1,0
12303  067100  001126  001714  001712
12304  067106  001736  001716  000000
12305  067114  001116  002032  001200  DT15:   .WORD   $ERRPC,TSTNM,$TMP1,0
12306  067122  000000
12307  067124  001116  002032  001204  DT16:   .WORD   $ERRPC,TSTNM,$TMP3,$TMP1,$TMP0,$BDDAT,0
12308  067132  001200  001176  001126
12309  067140  000000
12310  067142  001116  002032  001710  DT17:   .WORD   $ERRPC,TSTNM,BA,DB,WC,CS1,CS2,0
12311  067150  001704  001706  001714
12312  067156  001712  000000
12313
12314  067162  001116  002032  001716  DT20:   .WORD   $ERRPC,TSTNM,ER1,ER2,ER3,AS,DS1,0
12315  067170  001722  001730  001732
12316  067176  001736  000000
12317  067202  001116  002032  001714  DT21:   .WORD   $ERRPC,TSTNM,CS1,AS,DS1,0
12318  067210  001732  001736  000000
12319  067216  001116  002032  000000  DT22:   .WORD   $ERRPC,TSTNM,0
12320  067224  001116  002032  001720  DT24:   .WORD   $ERRPC,TSTNM,DST,$BDDAT,$TMP1,$TMP2,$TMP3,0
12321  067232  001126  001200  001202
12322  067240  001204  000000
12323  067244  001116  002032  002014  DT26:   .WORD   $ERRPC,TSTNM,PCJSR,$BDADR,CS1,CS2,DS1,ER1,0
12324  067252  001122  001714  001712
12325  067260  001736  001716  000000
12326  067266  001116  002032  002014  DT27:   .WORD   $ERRPC,TSTNM,PCJSR,REGADR,$GDDAT,$BDDAT,0
12327  067274  042204  001124  001126
12328  067302  000000
12329
12330  067304  001116  002032  002014  DT30:   .WORD   $ERRPC,TSTNM,PCJSR,REGADR,$GDDAT,$BDDAT,0
12331  067312  042204  001124  001126
12332  067320  000000
12333  067322  001116  002032  047320  DT31:   .WORD   $ERRPC,TSTNM,ERWORD,$GDDAT,$BDDAT,CS1,DS1,ER1,0
12334  067330  001124  001126  001714
12335  067336  001736  001716  000000
12336  067344  001116  002032  002014  DT32:   .WORD   $ERRPC,TSTNM,PCJSR,ERWORD,$GDDAT,$BDDAT,CS1,DS1,ER1,0
12337  067352  047320  001124  001126
12338  067360  001714  001736  001716
12339  067366  000000
12340  067370  001116  002032  002014  DT33:   .WORD   $ERRPC,TSTNM,PCJSR,ERWORD,$GDDAT,CS1,DS1,ER1,0
12341  067376  047320  001124  001714
12342  067404  001736  001716  000000
12343  067412  001116  002032  045230  DT34:   .WORD   $ERRPC,TSTNM,GECC1,GECC2,WECC1,WECC2,DISK,0
12344  067420  045232  052116  052120
12345  067426  051116  000000
12346  067432  001116  002032  045230  DT35:   .WORD   $ERRPC,TSTNM,GECC1,GECC2,EC2,EC1,POSITI,ER1,0
12347  067440  045232  001746  001744
12348  067446  045242  001716  000000
12349  067454  001116  002032  002014  DT36:   .WORD   $ERRPC,TSTNM,PCJSR,MR,EC1,EC2,0
```

```
12350  067462  001734  001744  001746
12351  067470  000000
12352  067472  001116  002032  001744  DT37:   .WORD   $ERRPC,TSTNM,EC1,POSITI,GECC1,GECC2,EC2,DATENV,ZCODE,0
12353  067500  045242  045230  045232
12354  067506  001746  045246  045250
12355  067514  000000
12356
12357  067516  001116  002032  001126  DT40:   .WORD   $ERRPC,TSTNM,$BDDAT,0
12358  067524  000000
12359
12360  067526     000     000     000  DF1:    .BYTE   0,0,0,0,0
12361  067531     000     000
12362  067533     000     000     001  DF2:    .BYTE   0,0,1,0
12363  067536     000
12364  067537     000     000     001  DF3:    .BYTE   0,0,1,0,0
12365  067542     000     000
12366
12367  067544     000     000     000  DF11:   .BYTE   0,0,0,0,0,0,0
12368  067547     000     000     000
12369  067552     000
12370  067553     000     000     000  DF14:   .BYTE   0,0,0,0,0,0,0,0
12371  067556     000     000     000
12372  067561     000     000
12373  067563     000     000     000  DF15:   .BYTE   0,0,0
12374  067566     000     000     000  DF16:   .BYTE   0,0,0,0,0
12375  067571     000     000
12376  067573     000     000     000  DF17:   .BYTE   0,0,0,0,0,0,0
12377  067576     000     000     000
12378  067601     000
12379
12380  067602     000     000     000  DF20:   .BYTE   0,0,0,0,0,0,0
12381  067605     000     000     000
12382  067610     000
12383  067611     000     000     000  DF21:   .BYTE   0,0,0,0,0
12384  067614     000     000
12385  067616     000     000     000  DF22:   .BYTE   0,0,0,0
12386  067621     000
12387  067622     000     000     000  DF24:   .BYTE   0,0,0,0,0,0,0
12388  067625     000     000     000
12389  067630     000
12390  067631     000     000     000  DF26:   .BYTE   0,0,0,0,0,0,0,0
12391  067634     000     000     000
12392  067637     000     000
12393  067641     000     000     000  DF27:   .BYTE   0,0,0,0,0,0
12394  067644     000     000     000
12395
12396  067647     000     000     000  DF30:   .BYTE   0,0,0,0,0,0
12397  067652     000     000     000
12398  067655     000     000     001  DF31:   .BYTE   0,0,1,0,0,0,0,0
12399  067660     000     000     000
12400  067663     000     000
12401  067665     000     000     000  DF32:   .BYTE   0,0,0,1,0,0,0,0,0
12402  067670     001     000     000
12403  067673     000     000     000
12404  067676     000     000     000  DF33:   .BYTE   0,0,0,1,0,0,0,0
12405  067701     001     000     000
```

```
12406  067704    000    000
12407  067706    000    000    000  DF34:   .BYTE   0,0,0,0,0,0,0
12408  067711    000    000    000
12409  067714    000
12410  067715    000    000    000  DF35:   .BYTE   0,0,0,0,0,0,0,0
12411  067720    000    000    000
12412  067723    000    000
12413  067725    000    000    000  DF36:   .BYTE   0,0,0,0,0,0
12414  067730    000    000    000
12415  067733    000    000    000  DF37:   .BYTE   0,0,0,0,0,0,0,0,0
12416  067736    000    000    000
12417  067741    000    000    000
12418
12419  067744    000    000    000  DF40:   .BYTE   0,0,0
12420          067750                        .EVEN
12421
12422          000001                .END
```

```
ABS    = 000200            918#
ACL    = 000040            971#
ACU    = 100000            907#
ADE    = 001000            823#   7923
AS       001732           1062#   3449    3517    6289    6347   12314   12317
ATA    = 100000            811#   3286    4882    6299    6319    7938    8085    8194    8307
ATABLE   004220           1156#   1602
ATTENT   002016           1094#   1561*   1602*   3721    4879    6277    6345    7936    8083    8192    8305
AT0    = 000001            845#
AT1    = 000002            846#
AT2    = 000004            847#
AT3    = 000010            848#
AT4    = 000020            849#
AT5    = 000040            850#
AT6    = 000100            851#
AT7    = 000200            852#
A16    = 000400            786#   3350    6536    6556
A17    = 001000            787#   3350    6536    6556
BA       001710           1051#   6690    6704   12310
BAI    = 000010            756#   3122    3127    6429    6548    6564    7039    9258
BASECH   046160           1324    9659#
BEGIN    004240            186    1169#  11554
BEGIN2   004230            188    1167#
BITST    042206           1359    1728    1753    1829    1853    1877    1959    1985    2010    2035    2056    8543#
BIT0   = 000001            146#   6284    9000    9035    9037
BIT00  = 000001            136#    146
BIT01  = 000002            135#    145
BIT02  = 000004            134#    144
BIT03  = 000010            133#    143
BIT04  = 000020            132#    142
BIT05  = 000040            131#    141    1936
BIT06  = 000100            130#    140    1936
BIT07  = 000200            129#    139    1936
BIT08  = 000400            128#    138    1937   10842
BIT09  = 001000            127#    137    1915    1936   10850   11330
BIT1   = 000002            145#   9014
BIT10  = 002000            126#  11314
BIT11  = 004000            125#  10857
BIT12  = 010000            124#
BIT13  = 020000            123#   9030    9032   11321
BIT14  = 040000            122#   9007   10928
BIT15  = 100000            121#   9002    9009    9016    9025    9027
BIT2   = 000004            144#
BIT3   = 000010            143#
BIT4   = 000020            142#
BIT5   = 000040            141#
BIT6   = 000100            140#
BIT7   = 000200            139#
BIT8   = 000400            138#
BIT9   = 001000            137#
BLT1     042234           8549#   8554
BLT2     042256           8549    8550    8556#
BLT3     042266           8557    8558#
BPTVEC= 000014              153#
CA       001726           1060#
CAT      036720           7686    7817#
```

```
CC      001752          1070#
CHECKD= 104414         11515#
CHECKS  042760           5015    5125    5249    5477    5604    5730    6850    8714#
CHECKT  042570           3810    4735    4985    5098    5220    5339    5446    5573    5699    5816    5910    6260    6425
                         6544    6819    6947    7023    7627    7757    7888    8056    8161    8266    8662#   8847    8947
                         9114   11515
CKSWR = 104407          10816   11308   11329   11510#
CLAREA  042452           5023    5133    5137    5257    5261    5783    5804    5874    5879    5927    5932    6001    6013
                         6020    6032    6069    6081    6142    6162    6207    6389    6394    6442    6511    6627    6647
                         6727    6919    6937    7145    7160    7226    7241    7308    7317    7385    7394    7446    7461
                         7515    7524    7594    7615    7724    7745    7837    7866    8608#
CLDISK  042534           1560    1680    1720    1772    1896    2089    2263    2514    2581    2686    2752    2823    2891
                         2934    2955    3020    3059    3104    3150    3183    3243    3260    3341    3418    3475    3554
                         3830    3971    4091    4227    4251    4370    4501    4631    4696    4719    4924    4956    5067
                         5187    5311    5434    5561    5687    5808    5899    6259    6342    6414    6532    6611    6807
                         6941    7022    7098    7109    7190    7271    7348    7566    7619    7698    7749    7836    7870
                         8009    8031    8118    8137    8227    8247    8340    8644#   8836    8935    9113    9301    9725
CLR   = 000040           758#    2282    2296    3918    4031    4162    4310    4440    4572    6285    7029    8649
COMHD   047040           5453    5580    5706    5822    5915    6431    6550    6876    6953    7633    7763    7904    8853
                         8952    9134   9822#
COMPA   050112           9970*  10013   10085#  10232*
COMPAR  043430           2161    2549    2644    3387    3455    3523    3898    3951    4014    4062    4145    4189    4292
                         4337    4422    4466    4554    4598    4670    4902    5031    5146    5269    5495    5622    5748
                         6089    6214    6734    6868    7082    7957    8101    8210    8323    8872#   9155    9181
COMWHD  052162           4995    5108    5230   10351#
COUNTD  050560          10226#  10230*  10247
CPHALT  062145           3811    4736    4986    5016    5099    5126    5221    5250    5340    5447    5478    5574    5605
                         5700    5731    5817    5911    6261    6426    6545    6820    6851    6948    7024    7628    7758
                         7899    8057    8162    8267    8848    8948    9115   11847#
CR    = 000015             61#  11002   11012
CRC     043742           4967    5077    5198    5322    5428    5555    5681    5798    5893    6408    6526    6801    6931
                         7609    7739    7860    8830    8929    8985#   9098
CRLF  = 000200             62#  10973   11012
CSF   = 000002            894#    912#
CSU   = 000010            896#    914#
CS1     001714           1055#   2700    2712    3314    5355    8665    8669    8675    8717    8724   12299   12302   12310
                        12317   12323   12333   12336   12340
CS2     001712           1052#   2692    2721    2757    2772    2833    2901    2939    2959    6046    6177    6179    6313
                         6667    6669   12299   12302   12310   12323
CYL     047200           5420*   5429    5547*   5556    5673*   5682    5790*   5799    5886*   5894    6401*   6409    6519*
                         6527    6793*   6802    6924*   7601*   7610    7731*   7740    7848*   7851*   7861    8823*   8831
                         8921*   8930    9082*   9089    9856#
DAREAD  047254           9896    9909#
DATENV  045246           9378#  10113*  10130*  10180*  10195*  12352
DAWORD  047260           5426*   5553*   5679*   5891*   6406*   6524*   6799*   8828*   8926*   9087*   9910#
DR      001704           1049#  12310
DCK   = 100000            829#    6887    9210    9215    9231    9234
DCL   = 000100            972#
DCLEAR  002050           1128#    3591
DDISP = 177570             68#     262    1200
DENVL = 000200            839#
DE1   = 000040            801#
DFF20 = 000002            797#
DF1     067526            317     436     443   12360#
DF11    067544            425     724   12367#
DF14    067553            456   12370#
```

```
DF15     067563            465   12373#
DF16     067566            476   12374#
DF17     067573            489   12376#
DF2      067533          12362#
DF20     067602            502   12380#
DF21     067611            513   12383#
DF22     067616            528   12385#
DF24     067622            549     562   12387#
DF26     067631            575   12390#
DF27     067641            590   12393#
DF3      067537            405     412   12364#
DF30     067647            603   12396#
DF31     067655            368     376   12398#
DF32     067665            352     393   12401#
DF33     067676            335   12404#
DF34     067706            621   12407#
DF35     067715            640     712   12410#
DF36     067725            655     672   12413#
DF37     067733            695   12415#
DF40     067744            733   12419#
DF5    = 000001            796#
DH1      062430            312     431   11886#
DH11     062676            417     716   11915#
DH14     063056            448   11934#
DH15     063256            462   11956#
DH16     063307            470   11961#
DH17     063452            481   11979#
DH2      062553            399   11900#
DH20     063635            494   11999#
DH21     064020            507   12019#
DH22     064140            523   12033#
DH24     064243            542     555   12045#
DH26     064426            566   12065#
DH27     064631            582   12087#
DH30     064774            596   12105#
DH31     065137            358   12123#
DH32     065342            341     382   12146#
DH33     065565            326   12172#
DH34     065770            611   12195#
DH35     066153            631     702   12215#
DH36     066356            646     663   12238#
DH37     066515            683   12255#
DH40     066733            729   12281#
DIGB   = 000004            793#
DISK     051116           2100    2108    2163    2177    5033    5148    5271    5405    5532    5658    5784    5834    5875
                          5876    6021    6022    6026    6033    6034    6150    6156    6163    6164    6390    6391    6512
                          6513    6635    6641    6648    6649    6654*   6778    6920    6963    7595    7645    7725    7775
                          7838    7979    9069    9183    9626    9640*   9641*   10253   10314#  10771   12343
DISPLA   001142             262#  1200*   1208*  10871*  11313*
DISPRE   000174             169#  1208
DLT    = 100000             763#  2690    2901    2989    2992
DL64   = 000020             800#
DMD    = 000001             833#  1900    1903    1924    2279    2516    2596    3244    3261    3353    3419    3476    3569
                          3590    3832    3920    3973    4033    4093    4164    4229    4246    4247    4253    4312    4372
                          4395    4396    4442    4503    4527    4528    4574    4633    4647    4651    4697    4750    4759
                          4881    4925    5349    7055    7059    7110    7191    7272    7349    7567    7699    7937    8010
```

```
                    8034    8076    8084    8140    8186    8193    8250    8299    8306    9265    9295    9564    9831
                    9974    9979   10093   10106   10109   10123   10126   10167   10177   10192   10360   10512   10515
                   10523   10533   10683   10692
DOG      036254     7554    7691#
DPR    = 000400      804#    3286    6299    6319    6355    7671    7674    7801    7804    8688    8691    8736
DRV    = 000200      803#    3286    3432    3500    3877    3928    3993    4041    4112    4171    4272    4320    4404
                    4449    4536    4581    6299    6319    6355    7061    7671    7674    7801    7804    8688    8696
                    8731
DST      001720     1057#   4767*   4838*  12320
DSWR   = 177570       67#     261    1199
DS1      001736     1064#    3284    6297    6317    6353    7669    7675    7799    7805    8686    8691    8696    8731
                    8736   12299   12302   12314   12317   12323   12333   12336   12340
DT       001740     1065#
DTAGAP   052122    10317#
DTE    = 010000      826#
DTSV   = 001000      841#    9995   10058   10061
DT1      067010      316     435     442   12292#
DT11     067052      424     723   12299#
DT14     067072      455   12302#
DT15     067114      464   12305#
DT16     067124      475   12307#
DT17     067142      488   12310#
DT2      067024    12294#
DT20     067162      501   12314#
DT21     067202      512   12317#
DT22     067216      527   12319#
DT24     067224      548     561   12320#
DT26     067244      574   12323#
DT27     067266      589   12326#
DT3      067036      404     411   12296#
DT30     067304      602   12330#
DT31     067322      367     375   12333#
DT32     067344      351     392   12336#
DT33     067370      334   12340#
DT34     067412      619   12343#
DT35     067432      638     710   12346#
DT36     067454      653     670   12349#
DT37     067472      693   12352#
DT40     067516      732   12357#
DVA    = 004000      789#    1784    1787    2615    2700    2712    3027    3029    3037    3039    3067    3069    3077
                    3079    6556    6558    8580    8665    8669    8724
ECDATA   045226     9359#    9407    9565*   9632*   9634*  10103*  10105*  10120*  10122*  10170*  10174*  10185*  10189*
ECH    = 000100      820#
ECI    = 004000      936#    3366    5440    5567    5693    5906    6421    6540    6813    8843    8943    9131
ECORR    045652     9559#
ECTEST   045264     9398#    9593    9635   10114   10131   10181   10196
EC1      001744     1067#   12346   12349   12352
EC2      001746     1068#   12346   12349   12352
EMTVEC=  000030      156#    1184*   1185*   1299*   1330*
EM1      057154      311     580   11571#
EM11     057325      416     429     564   11590#
EM14     057360      447   11595#
EM15     057400      460   11598#
EM16     057451      469   11605#
EM17     057472      480   11608#
EM2      057237      324     339     356   11580#
```

```
EM20    057516            493   11612#
EM21    057537            506   11615#
EM22    057561            517   11619#
EM24    060241            539   11673#
EM25    060334            552   11683#
EM30    060374            595   11689#
EM31    060516            608   11703#
EM32    060641            625   11718#
EM33    061130            645   11751#
EM34    061202            658   11758#
EM35    061275            677   11768#
EM36    061574            700   11803#
EM37    061705            715   11816#
EM40    062042            727   11834#
EM6     057266            380     398   11584#
ERCLFC  020504           3603    3614    3625    3639    3655    3665    3675    3685    3695    3705    3715    3725    3734
                         3745    3754    3765    3775    3785    3802#
ERCRC   052356          10381*  10430#  10604*
ERCS2C  013350           2308    2318    2330    2341    2351    2361    2371    2381    2391    2401    2410    2420    2431
                         2441    2453    2464    2475    2493#
ERFLGS  002006           1083#   2095*   2264*   2837*   2851    4992*   5013    5027*   5105*   5123    5142*   5227*   5247
                         5265*   5346*   5452*   5475    5579*   5602    5705*   5728    5821*   5831    5843    5914*   5925
                         5948    6087*   6193*   6430*   6441*   6461    6549*   6655*   6713*   6825*   6848    6952*   6960
                         6971    7632*   7642    7654    7762*   7772    7784    7893*   7972*   7987    8063*   8168*   8273*
                         8893    9133*   9144    9893   10464   11309*
ERHDGP  052360          10382*  10438#  10615*
ERHEAD  052354          10380*  10418#  10592*
ERPOS   045650           9555#   9561*   9578
ERR    = 040000            810#    3286    6299    6319    6355    7062    7938    8085    8194    8307
ERRVEC= 000004            149#    1197    1198*   1209*   1301*   1308*   1332*  10833   10834*  10836*  10839*
ERSTAR  047012           9725#
ERUNIT  047010           9724#   9726
ERWORD  047320           2557    2652    3395    3463    3531    3906    3960    4021    4071    4152    4198    4300    4346
                         4430    4476    4562    4607    4678    4910    5837*   5842*   5942*   5947*   6455*   6460*   6965*
                         6970*   7089    7648*   7653*   7778*   7783*   7965    7981*   7986*   8109    8218    8331    8886*
                         8892*   9961#   9997*  10017*  10021*  10043*  10063*  10542*  10549*  10553*  10575*  10583*  10589*
                        10603*  10608*  10612*  10624*  12294   12296   12333   12336   12340
ER1     001716           1056#   3265    3276    6363    6887    6990   12299   12302   12314   12323   12333   12336   12340
                        12346
ER2     001722           1058#  12314
ER3     001730           1061#  12314
EXT1   = 000001            874#
EXT10  = 000010            877#
EXT2   = 000002            875#
EXT20  = 000020            878#
EXT4   = 000004            876#
EXT40  = 000040            879#
FEN    = 000200            900#
FER    = 000020            818#    6887    6990
FILLEC  046024           9616#
FIRST   002034           1112#   1217    1258*
FMT22  = 010000            937#    3366    4725    4961    4977    4993    5071    5087    5106    5192    5209    5228    5316
                         5332    5347    5420    5440    5481    5547    5567    5608    5673    5693    5734    5790    5814
                         5886    5906    5997    6064    6401    6421    6519    6540    6813    6945    7040    7153    7234
                         7454    7601    7625    7731    7755    7848    7851    7876    8045    8064    8150    8169    8274
                         8823    8843    8921    8943    9082    9131    9259
```

```
FNWORD  052376          4966*  5076*  5197*  5321* 10468#
FORMAT  050562         10227#  10246
FOUT    052402         10465  10471#
FRMAT1  033700          6919#
FSYNER  052352         10379*  10405#  10578*
FUTABL  002042          1124#  3246
GCRC    052346          4969   5079   5200   5324   6933  10390#  10599  10602
GECC1   045230          9363#  9405   9491*  9624*  9640  12343  12346  12352
GECC2   045232          9366#  9406   9492*  9625*  9641  12343  12346  12352
GNS   = ****** U         168   1225   1230   1235   1239   1243   1247   1251   1255   1278   1317   1321   1403
                        1407   1411   1415   1443   1447   1451   1455   1505   1512   1518   1606   1612   1618
                        1647   1653   1659   1685   1689   1693   1699   4121   8355   8361   8462   8468   8475
                        8479   8483   8490   8494   9662   9668   9680   9686   9692   9696   9700   9706   9712
                       10794  11502  11503  11504  11505  11506  11508  11510  11511  11512  11513  11514  11515
                       11516
GO    = 000001           783#  2526   3274   3362   3429   3497   3592   3871   3875   3926   3987   3991   4039
                        4107   4110   4169   4267   4270   4317   4390   4402   4447   4522   4534   4579   4646
                        4751   5350   5352   6307   7063   8077   8187   8300   9266   9296   9834  10363
GRV   = 000010           799#
GTSWR = 104406          1265  11508#
HADTMP  045252          9388#  9580
HARDER  045244          9375#
HCCRCE  044344          7170   7251   7327   7404   7471   7534   9111#
HCE   = 000200           821#  9200   9204   9210   9215
HCI   = 002000           935#  3366
HCRC  = 000400           822#  9222   9225   9231   9234
HDESYN  052362         10383*  10451#  10627*
HDWSYN  051114         10312#
HEADER  051070         10309#
HEDGAP  047214          9849*  9871#  10024*
HEDSYN  047216          9850*  9882#  10044*  10065*
HEGAP   051102         10015  10311#
HT    = 000011            59#  10971  11012
IAE   = 002000           824#  8081   8190   8303
IE    = 000100           784#
ILF   = 000001           814#  3276
ILLEGL  002104          1142#  3257*  3262
ILR   = 000002           815#
IOTVEC= 000020           154#  1182*  1183*
IR    = 000100           759#  1794   2613   2690   2720   2755   2770   2937   2957   2992   3111   3117   3127
                        6044   6175   6564   6665   7921   8585
IXE   = 004000           904#   922#
KEY1    047204          5424*  5551*  5677*  5794*  5889*  6404*  6522*  6797*  6927*  7605*  7735*  7856*  8826*
                        8924*  9085*  9858#
KEY2    047206          5425*  5552*  5678*  5795*  5890*  6405*  6523*  6798*  6928*  7606*  7736*  7857*  8827*
                        8925*  9086*  9859#
KIPAR0= 172340           222#
KIPAR1= 172342           223#
KIPAR2= 172344           224#
KIPAR3= 172346           225#
KIPAR4= 172350           226#
KIPAR5= 172352           227#
KIPAR6= 172354           228#
KIPAR7= 172356           229#
KIPDR0= 172300           211#
KIPDR1= 172302           212#
```

```
KIPDR2= 172304          213#
KIPDR3= 172306          214#
KIPDR4= 172310          215#
KIPDR5= 172312          216#
KIPDR6= 172314          217#
KIPDR7= 172316          218#
LA      001750         1069#
LAD     042514         8557*  8626#  8630
LERR    042202         8540#  8546*  8587
LF    = 000012           60#  11006  11012
LST   = 002000          806#  7671   7674   7801   7804   793*
MAKECY  045106         4699   4926   7111   7192   7273   735*   7568   7700   8011   9287#
MASK    042200         8539#  8543*  8567*  8568   8569*  8572*  8573   8574*
MCLK  = 000002          834#  4790   4791   4792   4793   4819   4820   4821   4822   4868   4869   4872   4873
                       9591   9592   9599   9600   9976   9977   9979   9980   9982   9983   10098  10115  10167
                      10182  10515  10525  10710  10712  10713  10714  10715  10729  10730  10731  10732
MCPE  = 020000          790#
MHS   = 001000          902#   920#
MID     045076         9261*  9268#
MIDDLE  044774         3301   9250#
MINX  = 000004          835#  4234   4235   4377   4378   4510   4511   4758   8035   8036   8141   8142   8251
                       8252   9832   9833  10361  10362  10691
MMVEC = 000250          200#
MOL   = 010000          808#  1681   1695
MPE   = 000400          761#
MR      001734         1063#  12349
MRD   = 000020          837#  10096  10101  10109  10118  10126
MSE   = 000020          897#
MSTCK = 000010          836#  4246   4395   4527   4752   4753   4755   4756   4758   4790   4791   4819   4820
                       4854   4855   4857   4858   4872   4873   9975   9977   9979   9980   10098  10167  10514
                      10515  10684  10685  10686  10687  10691  10709  10712  10713  10729  10730  10739  10740
MWR   = 000040          838#  10168  10183  10516  10526
MXF   = 001000          762#  3311
NCODE   045236         9372#
NCOUNT  045240         9373#  9571*
NED   = 010000          765#  2110   9728
NEM   = 004000          764#  6564
NHS   = 002000          903#   921#
NOPERA  002042         1125#  3420   3488
NOSYNC  047314         6917*  7136*  7217*  7299*  7376*  7436*  7505*  9897   9958#  10032  10056  10817*
NOUNIT  001776         1076#  1472*  1526*  1533   1583*  8378*
NOWORD  047246         5796*  6929*  7607*  7737*  7858*  9905#
NUNIT   002000         1078#  1533*  1534*  3646
OCYL  = 100000          974#
OF      001724         1059#
OFREV = 000200          934#
OFSETC  002074         1138#  4257
OF100 = 000004          929#
OF200 = 000010          930#
OF25  = 000001          927#  4258   4318
OF400 = 000020          931#
OF50  = 000002          928#
OF800 = 000040          932#
OPERSE  041360         8458#  11054  11137
OPI   = 020000          827#
OR    = 000200          760#  2768   2770   2831   2954   2970   2992   6665   7920   8585
```

M7

```
OUT     047264          9894    9901    9907    9912#
PAR   = 000010           817#   1804    1807
PAT   = 000020           757#   1776    1794
PCJSR   002014          1092#   8662*   8663*   8714*   8715*   9111*   9112*   9289*   9290*   9559*   9560*   9822*   9823*
                       10351*  10352*  12323   12326   12330   12336   12340   12349
PCR   = 002000           763#   5355
PIE1  = 100000          9326#   9475    9490
PIE10 = 000100          9335#
PIE11 = 000040          9336#   9442
PIE12 = 000020          9337#   9475
PIE13 = 000010          9338#
PIE14 = 000004          9339#
PIE15 = 000002          9340#
PIE16 = 000001          9341#
PIE17 = 100000          9342#
PIE18 = 040000          9343#
PIE19 = 020000          9344#
PIE2  = 040000          9327#   9433
PIE20 = 010000          9345#
PIE21 = 004000          9346#   9450
PIE22 = 002000          9347#   9481
PIE23 = 001000          9348#   9458
PIE24 = 000400          9349#   9481
PIE25 = 000200          9350#
PIE26 = 000100          9351#
PIE27 = 000040          9352#
PIE28 = 000020          9353#
PIE29 = 000010          9354#
PIE3  = 020000          9328#   9475
PIE30 = 000004          9355#
PIE31 = 000002          9356#
PIE32 = 000001          9357#   9414    9422
PIE4  = 010000          9329#
PIE5  = 004000          9330#
PIE6  = 002000          9331#
PIE7  = 001000          9332#
PIE8  = 000400          9333#
PIE9  = 000200          9334#
PIP   = 020000           809#   3876    3927    3992    4040    4111    4170    4271    4319    4403    4448    4535    4580
PIRQ  = 177772            66#
PIRQVE= 000240           160#
PKACK   002100          1140#   2517
PLU   = 020000           906#    923#
POSITI  045242          9374#   9515    9569    9577*   9578    9590    9623*  12346   12352
PRBA    011170          1851*   1852#
PRCA    011642          2033*   2034#
PRCS1   010710          1751*   1752#
PRCS2   010644          1357    1726*   1727#
PRDST   011466          1957*   1958#
PRE   = 000020           970#
PRER1   011234          1875*   1876#
PRER2   011532          1983*   1984#
PRER3   011706          2054*   2055#
PROF    011576          2008*   2009#
PROG  = 001000           805#   2428    2625    3285    3742    6298    6318    6354    7670    7800    8687
PRWC    011124          1827*   1828#
```

A5

```
PR0    = 000000   83#
PR1    = 000040   84#
PR2    = 000100   85#
PR3    = 000140   86#
PR4    = 000200   87#
PR5    = 000240   88#
PR6    = 000300   89#
PR7    = 000340   90#
PS     = 177776   63#    64   1214*  8352*  8459*
PSEL   = 002000  788#
PSU    = 000001  967#
PSW    = 177776   64#
PUTREG  042140  2691   2711   2756   2771   2832   2899   2938   2958   3264   3275   3312   5354   5830
                6045   6176   6288   6308   6346   6666   6885   6959   7641   7771   7977   8514#  8664
                8716   9143   9512   9518   9586
PWRVEC= 000024   155#  1188*  1189* 11521* 11522* 11531* 11537* 11549* 11550*
P12     045256  9390#  9444*  9445*  9474
P22     045260  9391#  9452*  9453*  9479
P24     045262  9392#  9460*  9461*  9480
P3      045254  9389#  9435*  9436*  9473
RAW    = 000020   915#
RCOM    053520 10759# 10764*
RCYL    047304  9928#  9966* 10001
RDCHR  = 104410 11212  11511#
RDHEAD  047322  9851   9966#
RDLIN  = 104411 11259  11512#
RDOCT  = 104412  1280   8485   8496   9671   9688  11513#
RDY    = 000200   785#  1784   1787   2615   2700   2712   3027   3029   3037   3039   3067   3069   3077
                3079   3436   3504   6556   6558   7064   7684   7814   8580   8665   8675   8717
READ    050114  9990   9994  10002  10004  10006  10008  10010  10051  10055 10090#  10773  10781
READAT  002066 1135#  5900   6415   6533
READIN  002102 1141#  3351
RECALI  002046 1127#  4513
REDATA  053522  9909 10763#
REFOR   002070 1136#  5451   5578   5704   6824   8061
REGADR  042204 1384*  1788*  1799*  1808*  1908*  1930*  2289*  2493*  2559*  2654*  2785*  2792*  2850*
               2909*  2974*  2982*  2995*  3030*  3040*  3070*  3080*  3118*  3133*  3160*  3168*  3193*
               3201*  3397*  3465*  3533*  3802*  3908*  3962*  4023*  4073*  4154*  4200*  4302*  4348*
               4432*  4478*  4564*  4609*  4680*  4912*  6311*  6559*  6569*  6577*  7091*  7673*  7803*
               7967*  8111*  8220*  8333*  8541#  8545*  9203*  9214*  9224*  9235*  9306* 12292  12326
              12330
REGSA1  056142  1299 11310#
REINTO  003154 1149#  2522   2529*  2543   2550   2602   2611*  2614*  2615*  2616*  2617*  2618*  2619*
               2620*  2621*  2624   2628*  2629*  2630*  2641   2645   3358   3365*  3366*  3368*  3369*
               3382   3388   3867   3875*  3876*  3877*  3893   3899   3926*  3927*  3928*  3929*  3930*
               3946   3952   3983   3991*  3992*  3993*  4008   4015   4039*  4040*  4041*  4042*  4043*
               4057   4063   4103   4110*  4111*  4112*  4139   4146   4169*  4170*  4171*  4172*  4186
               4190   4263   4270*  4271*  4272*  4286   4293   4317*  4318*  4319*  4320*  4321*  4334
               4338   4386   4402*  4403*  4404*  4417   4423   4447*  4448*  4449*  4450*  4462   4467
               4518   4534*  4535*  4536*  4549   4555   4579*  4580*  4581*  4582*  4594   4599   4642
               4651*  4652*  4665   4671   4732   4878*  4879*  4881*  4882*  4883*  4897   4903   5024
               5025   5032   5134   5135   5138   5139   5147   5258   5259   5262   5263   5270   5436
               5497   5563   5624   5689   5750   5880   5881   5902   5940   5996   6002   6003   6007
               6014   6015   6091   6131   6136   6143   6144   6216   6267   6281   6395   6396   6417
               6453   6616   6621   6628   6629   6690   6704   6736   6809   6870   7047   7059*  7060*
               7061*  7062*  7063*  7064*  7077   7083   7146   7147   7176   7227   7228   7257   7318
```

B5

```
                    7319    7395    7396    7447    7448    7477    7525    7526    7899    7911*   7912*   7916*   7917*
                    7920*   7921*   7922*   7923*   7924*   7932*   7934*   7936*   7937*   7938*   7952    7958    8042
                    8070    8080*   8081*   8082*   8083*   8084*   8085*   8096    8102    8180    8190*   8191*   8192*
                    8193*   8194*   8205    8211    8293    8303*   8304*   8305*   8306*   8307*   8318    8324    8839
                    8939    9157    9182
RELEAS  002052      1129#   4637
RESVEC= 000010       150#
RETCL   002076      1139#   4381
RHAS    001656      1028#   1378    1418    6276    6345*
RHBA    001634      1011#   1851    2270*   2587*   3184    3346*   3560*   4975*   5085*   5207*   5330*   5436*   5563*
                    5689*   5810*   5902*   6417*   6535*   6572    6577    6809*   6943*   7037*   7621*   7751*   7872*
                    8043*   8148*   8258*   8839*   8939*   9128*   9256*
RHBAE   001700      1040#
RHCA    001652      1026#   2033    2277*   2594*   3349*   3567*   3852*   3861*   3978*   4240*   4724*   4994*   5107*
                    5229*   5348*   5444*   5571*   5697*   5815*   5909*   6424*   6543*   6817*   6946*   7027    7041*
                    7626*   7756*   7883*   7885*   8065*   8170*   8279*   8292*   8846*   8946*   9120*   9260*   9292*
RHCC    001676      1036#   3837    4240    8520    9302    9306
RHCS1   001640      1021#   1751    2118    2272*   2517*   2526*   2589*   3352*   3362*   3562*   3836*   3871*   3977*
                    3987*   4097*   4107*   4257*   4267*   4381*   4390*   4513*   4522*   4637*   4646*   4751*   5350*
                    5352*   6307*   7034*   7046    7056*   7070    7091    7681    7811    8077*   8187*   8300*   8578
                    8644    9253*   9266*   9294*   9296*   9664    9702    9834*   10363*
RHCS2   001636      1012#   1358    1419    1478*   1590*   1726    1727*   1752*   1776*   1828*   1852*   1876*   1958*
                    1984*   2009*   2034*   2055*   2271*   2588*   2608*   2765    2951    3484*   3561*   5357    6285*
                    6286*   6429*   6548*   7029*   7030*   7039*   8563*   8582    8645    9258*
RHCS3   001702      1041#
RHDB    001630      1009#   1303    2268*   2284    2688    2763*   2764*   2778    2779    2785    2792    2826*   2840
                    2850    2895*   2905    2909    2944    2964    3558*   3593    9672
RHDST   001644      1023#   1957    2274*   2591*   3347*   3564*   3835*   4722*   4747    4767    4838    4982*   5096*
                    5218*   5337*   5439*   5566*   5692*   5813*   5905*   6420*   6539*   6812*   6944*   7038*   7624*
                    7754*   7875*   7924    8054*   8159*   8264*   8842*   8942*   9125*   9257*   9293*
RHDS1   001662      1030#   3293    3427    3495    7673    7803    8646
RHDT    001664      1031#   1479    1481    1487    1489    1492    1494    1514    1592    1594    1620    1624    1626
                    1629    1631    1634    1636    1664    1666    2130    2132    2136    2138    2141    2143
RHEC1   001670      1033#   9515
RHEC2   001672      1034#   3423    3482    3491    9509
RHER1   001642      1022#   1422    1778*   1875    2273*   2590*   3563*   6269    8179    8198    8220    8292    8311
                    8333    8647    9199    9203    9209    9214    9221    9224    9229    9235
RHER2   001646      1024#   1983    2275*   2592*   3565*   6271
RHER3   001654      1027#   2054    2278*   2595*   3568*   6273
RHLA    001674      1035#   3929    4042    4172    4321    4450    4582    4652    4766    4836
RHMR    001660      1029#   1897    2279*   2280*   2516*   2596*   2597*   3244*   3261*   3353*   3419*   3476*   3569*
                    3570*   3590*   3832*   3920*   3973*   4033*   4093*   4164*   4229*   4234*   4235*   4246*   4247*
                    4253*   4312*   4372*   4377*   4378*   4395*   4396*   4442*   4503*   4510*   4511*   4527*   4528*
                    4574*   4633*   4647*   4697*   4746    4925*   5349*   7055*   7110*   7191*   7272*   7349*   7567*
                    7699*   8010*   8034*   8035*   8036*   8075    8140*   8141*   8142*   8185    8250*   8251*   8252*
                    8298    9265*   9295*   9563    9831*   9832*   9833*   9972    10245   10360*  10361*  10362*  10511
                   10681
RHOF    001650      1025#   2008    2276*   2593*   3348*   3566*   4258*   4725*   4993*   5106*   5228*   5347*   5440*
                    5567*   5693*   5814*   5906*   6421*   6540*   6813*   6945*   7040*   7625*   7755*   7876*   8064*
                    8169*   8274*   8843*   8943*   9131*   9259*
RHSN    001666      1032#   1614    1663    1665
RHWC    001632      1010#   1827    2269*   2521    2536    2559    2586*   2601    2634    2654    3151    3345*   3357
                    3375    3397    3421    3423    3442    3465    3481    3482    3489    3491    3510    3533    3559*
                    3866    3885    3908    3937    3962    3982    4000    4023    4050    4073    4102    4132    4154
                    4178    4200    4262    4279    4302    4327    4348    4385    4410    4432    4455    4478    4517
                    4542    4564    4587    4609    4641    4658    4680    4731    4890    4912    4973*   5008    5083*
```

CR

```
                    5118    5204*   5240    5328*   5435*   5465    5562*   5592    5688*   5718    5809*   5901*   6416*
                    6534*   6808*   6838    6942*   7036*   7620*   7750*   7871*   7898    7944    7967    8041*   8069
                    8089    8111    8146*   8256*   8518    8520    8838*   8938*   9126*   9255*
RH70     002040     1116#   1272*   1351    1715    2681    2747    2812    2886    2929    3014    3099    6505    6680
                    7908
RH70CK   005364     1270#
RKEY1    047310     9930#   9968*   10005
RKEY2    047312     9931#   9969*   10007
RMR    = 000004      816#   7060
RNCTR1   052160    10350#  10365*  10367*
RNO      053516    10758#  10763*
RNWAT1   052240    10365#
RPVEC    001626      999#   1215*   9682    9689*   9708    10797*
RPVECT   053632     1215   10791#  10797
RP06     002036     1114#   1591*   1597*   3843    7551    7845    7881    7930    8276
RSETR    047306     9929#   9967*   10003
RSYNC    047302     9927#   9993    9998    10030   10041   10054   10064   10573   10576   10622   10626
RUNCTR   047036     9821#   9836*   9838*
RUNWAT   047116     9836#
SAVDT    002010     1085#   1666*   2437    2439    3750    3752
SAVER    043226     2520    2535    2600    2633    3356    3374    3441    3509    3865    3884    3936    3981    3999
                    4049    4101    4131    4177    4261    4278    4326    4384    4409    4454    4516    4541    4586
                    4640    4657    4729    4889    7045    7069    7897    7943    8068    8088    8178    8197    8291
                    8310    8799#
SAVSN    002012     1088#   1665*   2449    2451    3761    3763
SC     = 100000      792#   1784    1787    2700    4878    6556    6558    7922    8080
SCOP1  = 104413    11514#
SCRC     052430    10501#  10508*
SCYL     052420    10497#  10504*  10582
SC1    = 000100      880#
SC10   = 001000      883#
SC2    = 000200      881#
SC20   = 002000      884#
SC4    = 000400      882#
SEARCH   053246     9267    9844    10374   10674#
SECGAP   051020     4951    5062    5182    5306    10307#  10510
SECTR    047202     5422*   5423*   5549*   5550*   5675*   5676*   5792*   5793*   5887*   5888*   6402*   6403*   6520*
                    6521*   6795*   6796*   6925*   6926*   7603*   7604*   7733*   7734*   7854*   7855*   8824*   8825*
                    8922*   8923*   9083*   9084*   9841    9857#
SECTR    053244     4749*   4771    4841    10672#  10674*  10682
SEECOM   002072     1137#   3836    3977    4241    9294
SEGPER   052350    10378*  10395#  10545*  10556*
SELECT   002002     1080#   1167*   1169*   1274    1537    1570    8367
SELTST   007444     1437    1537#
SERCH    002054     1130#   4726
SETDSK   044216     7141    7222    7304    7381    7441    7510    9063#
SILOTB   051116     2818    2838    2843    10313#
SKEY1    052424    10499#  10506*
SKEY2    052426    10500#  10507*
SKI    = 040000      973#
SN       001742     1066#
SND1     005326     1221    1258#
SPACE2   062425     2193   11883#
SPACE8   062417     2187    2188    2196    2197    2198   11881#
SR0    = 177572      204#
SR1    = 177574      205#
```

```
SR2    = 177576            206#
SR3    = 172516            207#
SSECTR  052422            10498#  10505*
SSYN    047212             9848*   9863#   9999*
STACK  = 001000              54#   1180    1296    1346    1723    1748    1770    1824    1848    1872    1954    1980    2005
                            2030    2051    2086    2511    2579    2676    2742    3019    3058    3149    3181    3241    3338
                            3552    3828    4089    4225    4368    4499    4629    4694    4717    4922    4948    5059    5179
                            5303    5396    5523    5649    5780    5871    5992    6125    6257    6386    6500    6609    6769
                            6913    7020    7107    7134    7188    7215    7269    7296    7346    7373    7434    7502    7564
                            7591    7696    7721    7834    8007    8029    8135    8245
START   004244              1168    1172#
STARTA  004466              1214#
STKLMT= 177774                65#
SWR     001140               261#   1178    1199*   1201    1207*   1263    1270    1312    1398    1435    2183    2864    4116
                            4208    5850    5964    6477    6981    7661    7791    7993    8558    8628    8899    9500    9502
                           10828   10842   10844   10850   10857   11057   11096   11151*  11314   11321   11326   11330   11529
                           11542*
SWREG   000176              170#   1207    1263   11057   11096   11119
SW0    = 000001             118#
SW00   = 000001             109#    118
SW01   = 000002             107#    117
SW02   = 000004             106#    116
SW03   = 000010             105#    115
SW04   = 000020             104#    114
SW05   = 000040             103#    113
SW06   = 000100             102#    112
SW07   = 000200             101#    111    2865    2866    5852    5966    6479    6983    7663    7793    7995    8901
SW08   = 000400             100#    110    2865
SW09   = 001000              99#    109    8558    8628
SW1    = 000002             117#
SW10   = 002000              98#
SW11   = 004000              97#
SW12   = 010000              96#   1270
SW13   = 020000              95#   1312    1398    1435    2183    4116
SW14   = 040000              94#
SW15   = 100000              93#
SW2    = 000004             116#
SW3    = 000010             115#
SW4    = 000020             114#   4208
SW5    = 000040             113#
SW6    = 000100             112#   9502
SW7    = 000200             111#
SW8    = 000400             110#   9500
SW9    = 001000             109#
TAGDTE  002030              1107#
TBITVE= 000014               151#
TDF    = 000040              898#    916#
TESDTE  002026              1103#  10011   10568   10826*
TESTAD  041356              8456#
TIMCNT  043104              8752#   8761    8766
TKVEC  = 000060               158#  11034*  11035*
TMPILL  002022              1097#   3245*   3248    3250*   3255    3257    3258*
TOLGAP  052124              5785    6921    7596    7726    7839   10241   10318#
TOTALA  002020              1095#   1428*   1430*   3575    3720
TPVEC  = 000064               159#
TRAPVE= 000034               157#   1186*   1187*
```

```
TRE    = 040000            791#   2700    2709    3314    6556    6558    7922    8080
TRK      047152           9843*   9846#
TRK1   = 004000            885#
TRK10  = 040000            888#
TRK2   = 010000            886#
TRK20  = 100000            889#
TRK4   = 020000            887#
TRTVEC= 000014             152#
TSECC    002024           1099#  10250   10766   10776   10818*
TSECCG   045234           9369#   9493   10099   10111   10116   10128   10178   10193   10252*  10264*  10768*  10778*  10822*
TSTNM    002032           1110#   1297*   1347*   1376*   1396*   1559*   1679*   1711*   1724*   1749*   1771*   1825*   1849*
                          1873*   1895*   1955*   1981*   2006*   2031*   2052*   2087*   2512*   2580*   2677*   2743*   2808*
                          2882*   2925*   3010*   3057*   3095*   3148*   3182*   3242*   3339*   3415*   3553*   3829*   4090*
                          4226*   4369*   4500*   4630*   4695*   4718*   4923*   4949*   5060*   5180*   5304*   5397*   5524*
                          5650*   5777*   5781*   5873*   5993*   6126*   6258*   6388*   6501*   6610*   6771*   6915*   7021*
                          7109*   7135*   7189*   7216*   7270*   7297*   7347*   7374*   7435*   7503*   7565*   7592*   7697*
                          7722*   7835*   8008*   8030*   8136*   8246*   8464   12292   12294   12296   12299   12302   12305
                         12307   12310   12314   12317   12319   12320   12323   12326   12330   12333   12336   12340   12343
                         12346   12349   12352   12357
TST1     005500           1275    1294#   9427
TST10    010666           1717    1746#
TST100   034512           7106#
TST101   034546           7131#
TST102   034670           7187#
TST103   034724           7212#
TST104   035046           7268#
TST105   035102           7293#
TST106   035202           7345#
TST107   035236           7370#
TST11    010732           1768#
TST110   035336           7431#
TST111   035464           7499#
TST112   035604           7563#
TST113   035640           7588#
TST114   036254           7695#
TST115   036310           7718#
TST116   036720           7833#
TST117   037546           8006#
TST12    011102           1822#
TST120   037602           9028#
TST121   040134           9134#
TST122   040460           8244#
TST123   041010           8350#
TST13    011146           1846#
TST14    011212           1870#
TST15    011256           1893#
TST16    011444           1951#
TST17    011510           1977#
TST2     005754           1344#
TST20    011554           2003#
TST21    011620           2023#
TST22    011664           2049#
TST23    011730           2085#
TST24    012336           2261#
TST25    013362           2490    2510#
TST26    013524           2578#
```

```
TST27   014074          2675#
TST3    006040          1353    1373#
TST30   014252          2683    2722    2741#
TST31   014500          2749    2789    2807#
TST32   014740          2814    2881#
TST33   015064          2888    2908    2923#
TST34   015364          2931    2990    3008#
TST35   015522          3016    3055#
TST36   015642          3039    3045    3078    3093#
TST37   016020          3101    3132    3146#
TST4    006112          1382    1392#
TST40   016144          3166    3179#
TST41   016270          3199    3239#
TST42   016564          3337#
TST43   017010          3414#
TST44   017400          3551#
TST45   020532          3800    3827#
TST46   021450          4088#
TST47   022124          4224#
TST5    007460          1538    1555#   9376    9396
TST50   022532          4367#
TST51   023076          4210    4498#
TST52   023442          4628#
TST53   023620          4693#
TST54   023654          4716#
TST55   024542          4921#
TST56   024576          4947#
TST57   025106          5014    5037    5058#
TST6    010232          1678#
TST60   025444          5124    5152    5178#
TST61   026004          5248    5275    5302#
TST62   026250          5395#
TST63   026634          5476    5501    5522#
TST64   027220          5603    5628    5648#
TST65   027604          5729    5754    5775#
TST66   030144          5832    5853    5870#
TST67   030510          5926    5967    5985#
TST7    010576          1682    1710#
TST70   031040          6095    6117#
TST71   031346          6221    6255#
TST72   031736          6365    6395#
TST73   032302          6480    6499#
TST74   032646          6507    6601#
TST75   033254          6741    6766#
TST76   033656          6849    6888    6909#
TST77   034206          6991    7018#
TTND  = 000122          1297#   1347#   1376#   1396#   1559#   1679#   1711#   1724#   1749#   1771#   1825#   1849#   1873#
                        1895#   1955#   1981#   2006#   2031#   2052#   2087#   2512#   2580#   2677#   2743#   2808#   2882#
                        2925#   3010#   3057#   3095#   3148#   3182#   3242#   3339#   3415#   3553#   3829#   4090#   4226#
                        4369#   4500#   4630#   4695#   4718#   4923#   4949#   5060#   5180#   5304#   5397#   5524#   5650#
                        5777#   5781#   5873#   5993#   6126#   6258#   6388#   6501#   6610#   6771#   6915#   7021#   7108#
                        7135#   7189#   7216#   7270#   7297#   7347#   7374#   7435#   7503#   7565#   7592#   7697#   7722#
                        7835#   8008#   8030#   8136#   8246#
TUF   = 000100           899#    917#
TY      047316          9960#
TYPDS = 104405          1509    1525    1609    4125    8358    8364    8374    8416    11382   11506#
```

```
TYPE   = 104401          1223   1278   1233   1237   1241   1245   1249   1253   1276   1315   1319   1401   1405
                         1409   1413   1441   1445   1449   1453   1503   1510   1516   1523   1604   1610   1616
                         1645   1651   1657   1683   1687   1691   1697   2197   2188   2193   2196   2197   2198
                         2207   3811   4119   4736   4986   5016   5099   5176   5221   5250   5340   5447   5478
                         5574   5605   5700   5731   5817   5911   6261   6426   6545   6820   6851   6948   7024
                         7628   7758   7889   8057   8162   8267   8353   8359   8372   8375   8414   8417   8460
                         8466   8472   8473   8477   8481   8488   8492   8848   8948   9115   9660   9666   9678
                         9684   9690   9694   9698   9704   9710   10792  10933  10976  11051  11063  11117  11118
                        11121  11132  11142  11153  11172  11216  11219  11223  11288  11290  11316  11324  11346
                        11363  11365  11368  11370  11385  11390  11454  11502#  11551
TYPOC  = 104402          1326   1515   1615   1621   8465   8471   9665   9683   9703   9709   10796  11120  11354
                        11378  11503#
TYPON  = 104404         11505#
TYPOS  = 104403          2190   2203   11504#
T.SCOP   042516          8628#  11514
UN       006030          1357*  1358*  1360#
UNIT     001774          1075#  1282*  1532*  1539*  1565   1582*  1584   1608   1727   1752   1793   1828   1852
                         1876   1958   1984   2009   2034   2055   2283   2299   2325   2608   2612   2689   2719
                         2754   2769   2830   2936   2956   2991   3110   3116   3121   3126   3484   3585   3634
                         3919   4032   4124   4163   4311   4441   4573   6043   6174   6286   6563   6664   7030
                         8357   8380   8385*  8563   8650
UNITS    001754          1074#  1465   1470   1532   1577   2162   2176   8381
UNITSL   002004          1081#  1283*  1539
UNLOAD   002044          1126#  4097
UNS    = 040000           828#
UPE    = 020000           766#  3123   3127
US1    = 000001           753#
US2    = 000002           754#
US4    = 000004           755#
UWR    = 000010           969#
VUF    = 000002           969#
VU30   = 010000           905#
VV     = 000100           802#  2428   2529   3286   3368   6298   6318   6355   7671   7674   7801   7804   8687
WAIT.T   043106          8753#  11516
WAT    = 104415          2766   2952   3309   3430   3433   3498   3501   7682   7812   11516#
WC       001706          1050#  3422   3456   3490   3524   6683   6697   8519   12310
WCE    = 040000           767#  6048   6179   6665   6669
WCF    = 000040           819#
WCRC     051100          5430   5557   5683   5800   5895   6410   6528   6803   7442*  7511*  7611   7741   7862
                         8832   8931   9090   10009  10310#
WCU    = 000001           893#   911#
WCYL     052336          4961*  4968   5071*  5078   5192*  5199   5316*  5323   6932   10386#
WECC1    052116          5018*  5128*  5252*  10315#  12343
WECC2    052120          5019*  5129*  5253*  10316#  12343
WKEY1    052342          4964*  5074*  5195*  5319*  10388#
WKEY2    052344          4965*  5075*  5196*  5320*  10389#
WLE    = 004000           825#
WORD     050110          9989*  9993*  10001*  10003*  10005*  10007*  10009*  10050*  10054*  10084#  10094*  10107*  10124*
                        10772*  10780*
WRCHDA   043572          6167   6656   8921#
WRCHDT   002060          1132#  8851
WRCHEK   002056          1131#  8951
WRCHHD   043260          6037   8823#
WRDATA   050566          9903   10229#  10467
WRFROM   002110          1148#  2537   2543*  2551   2635   2641*  2646   3376   3382*  3389   3443   3449*  3457
                         3511   3517*  3525   3886   3893*  3900   3938   3946*  3953   4001   4008*  4016   4051
```

HR

```
                    4057*  4064   4133   4139*  4147   4179   4186*  4191   4280   4286*  4294   4328   4334*
                    4339   4411   4417*  4424   4456   4462*  4468   4543   4549*  4556   4588   4594*  4600
                    4659   4665*  4672   4891   4897*  4905   4974   5084   5205   5329   5480   5496   5607
                    5623   5733   5749   5805   5806   5810   5928   5929   5933   5934   5939   6063   6070
                    6071   6075   6082   6083   6090   6195   6201   6208   6209   6215   6443   6444   6448*
                    6452   6715   6721   6728   6729   6735   6853   6869   6938   6939   6943   7037   7071
                    7077*  7084   7152   7161   7162   7233   7242   7243   7309   7310   7333   7386   7387
                    7410   7453   7462   7463   7516   7517   7540   7616   7617   7621   7746   7747   7751
                    7867   7868   7872   7912   7917   7945   7952*  7959   8090   8096*  8103   8147   8199
                    8205*  8212   8257   8312   8318*  8325   9156   9256
WRHEAD  052432      10385  10504#
WRIDAT  002062       1133#  5820   6951   7631   7761   7892   8271
WRIFOR  002064       1134#  4990   5103   5225   5344   7034   8166   9253
WRITE   050346      10018  10029  10158#  10254  10260  10266  10270  10273  10550  10570  10584  10597  10609
                    10620
WRL   = 004000        807#
WRU   = 000400        901#   919#
WSECTR  052340       4963*  5073*  5194*  5318*  10370  10387#
WSSYNC  051066      10308#
WSU   = 000004        895#   913#
WTRK    052274      10372*  10375#
WWORD   050344      10019  10023  10026  10030  10037  10042  10046  10153#  10202*  10255  10261  10267  10271
                    10274  10551  10555  10572  10585  10598  10610  10621
X       047210       5427*  5554*  5680*  5797*  5892*  6407*  6525*  6800*  6930*  7608*  7738*  7859*  8829*
                     8927*  9129*  9146   9860#  9895
XE2     007106       1433   1462#
Y       047250       9906#
ZCODE   045250       9381#  9574*  12352
ZER   = 000400        840#   9583
ZWORDS  050564      10228#  10248*
$AUTOB  001134        258#   1267*  11110  11239
$BDADR  001122        253#   2694*  2703*  2715*  2723*  2759*  2774*  2835*  2903*  2941*  2961*  3267*  3278*
                     3293*  3317*  5357*  8671*  8677*  8681*  8693*  8698*  8701*  8720*  8727*  8733*  8738*
                    12299  12302  12323
$BDDAT  001126        255#   1380*  1381   1783*  1784   1796*  1797   1803*  1804   1907*  1929*  2180*  2200
                     2202   2290*  2291   2305*  2306   2315*  2316   2327*  2328   2338*  2339   2348*  2349
                     2358*  2359   2368*  2369   2378*  2379   2388*  2389   2398*  2399   2406*  2408   2417*
                     2418   2427*  2438*  2439   2450*  2451   2461*  2462   2472*  2473   2784*  2791*  2848*
                     2905*  2907   2976*  2984*  2994*  3026*  3027   3036*  3037   3065*  3067   3076*  3077
                     3112*  3113   3129*  3130   3156*  3157   3164*  3165   3189*  3190   3197*  3198   3600*
                     3601   3611*  3612   3622*  3623   3636*  3637   3651*  3653   3662*  3663   3672*  3673
                     3682*  3683   3692*  3693   3702*  3703   3712*  3713   3722*  3723   3731*  3732   3741*
                     3751*  3752   3762*  3763   3772*  3773   3782*  3783   4766*  4768   4836*  4839   5008*
                     5118*  5240*  5465*  5592*  5718*  5841*  5946*  6313*  6459*  6554*  6556   6566*  6567
                     6572*  6574   6838*  6969*  7652*  7675*  7782*  7805*  7985*  8571*  8573*  8575   8577*
                     8771*  8891*  9199*  9200   9209*  9210   9221*  9222   9229*  9231   9302*  9303   10023*
                    10042*  10544*  10555*  10577*  10591*  10601*  10614*  10625*  12292  12296  12302  12307  12320
                    12326  12330  12333  12336  12357
$BELL   001716        287#   11063  11232  11316  11338
$CHARC  054646      10978*  10988*  10995  11004*  11009#
$CKSWR  055160      11096#  11510
$CMTAG  001100        241#   1175   1176   1184   1190   1191
$CM1  = 000006        273#    274#   275#   276#   277#   278#   279#
$CM2  = 000014        273#    274#   275#   276#   277#   278#   279#
$CM3  = 000006        271#    273
$CM4  = 000006        279#    280#   281#   282#   283#   284#   285#
```

```
$CNTLC  055731          11051   11132   11232#
$CNTLG  055743          11117   11234#
$CNTLU  055736          11142   11233#
$CRLF   001223            289#   1523    2207    8472   10977   11012   11153   11232   11293   11324   11338   11346   11365
                        11370   11390
$DBLK   054422          10899   10933   10941#
$DOAGN  041332           8410    8419    8425#
$DTBL   054412          10902   10937#
$ENDAD  041322            178    8421#
$ENDCT  041270           8412#
$ENDMG  041341           8372    8414    8429#
$ENULL  041336           8375    8417    8428#
$EOP    041234           1328    1458    8379    8402#
$EOPCT  041262           8409#   8413
$ERFLG  001103            244#   8560   10806   10846   10848   10854*  10875   11311*  11338
$ERMAX  001115            250#   1192*  10848   10870*  10875
$ERROR  056132           1184    1330   11307#
$ERRPC  001116            251#  11318*  11319*  11320   11338   11352   12292   12294   12296   12299   12302   12305   12307
                        12310   12314   12317   12319   12320   12323   12326   12330   12333   12336   12340   12343   12346
                        12349   12352   12357
$ERRTB  001226            306#  11360
$ERRTY  056300          11323   11345#
$ERTTL  001112            248#   8363    8365*  11317*  11338
$ESCAP  001214            286#   1191*  10869*  11333   11335   11338
$FILLC  001156            269#  10981   11012
$FILLS  001155            268#  11012
$GDADR  001120            252#
$GDDAT  001124            254#   1383*   1787*   1795*   1797    1807*   1904*   1928*   2178*   2185    2189    2288*   2291
                         2304*   2314*   2326*   2328    2337*   2347*   2357*   2367*   2377*   2387*   2397*   2407*   2408
                         2416*   2425*   2437*   2449*   2460*   2471*   2782*   2790*   2849*   2906*   2907    2975*   2983*
                         2993*   3029*   3039*   3069*   3079*   3116*   3117*   3128*   3159*   3167*   3192*   3200*   3599*
                         3610*   3621*   3635*   3637    3648*   3650*   3653    3661*   3671*   3681*   3691*   3701*   3711*
                         3720*   3721*   3723    3730*   3739*   3750*   3761*   3771*   3781*   5840*   5945*   6312*   6458*
                         6558*   6565*   6567    6576*   6968*   7651*   7674*   7781*   7804*   7984*   8566*   8568*   8570
                         8575    8580*   8586*   8890*   9204*   9215*   9225*   9234*   9305*   9998*  10022*  10039*  10041*
                        10064*  10543*  10554*  10576*  10590*  10602*  10613*  10626*  12292   12294   12296   12326   12330
                        12333   12336   12340
$GET42  041312           9418#
$GTSWR  055250          11118#  11508
$HD   = 000000             11
$HIOCT  056130          11281*  11292#
$ICNT   001104            245#  10861*  10862   10864*  10874
$ILLUP  057136          11521   11537   11556#
$INTAG  001135            259#  11115*  11134   11154   11239
$ITEMP  001114            249#  11320*  11338   11349
$LF     001224            290#  11012   11223   11232   11293   11338
$LPADR  001106            246#   1193*   1557*   8487*   8498   10852*  10867*  10872   10874
$LPERR  001110            247#   1194*   3263*   3474*   6278*   6343*   8470    8497*  10852   10868*  10874   11332
$MAIL = ****** U         1211    1263   10867   10965   11326
$MNEW   055761          11121   11237#
$MSWR   055750          11119   11235#
$MXCNT  054204          10865   10874#
$NULL   001154            267#  10983   11012
$NWTST= 000001           1288#   1290    1337#   1339    1370#   1389#   1543#   1545    1670#   1672    1704#   1706    1740#
                         1742    1762#   1764    1816#   1818    1840#   1842    1864#   1866    1887#   1889    1945#   1947
                         1971#   1973    1997#   1999    2022#   2024    2043#   2045    2077#   2079    2250#   2252    2501#
```

```
                    2503    2570#   2572    2665#   2667    2727#   2729    2799#   2801    2875#   2877    2915#   2917
                    3002#   3004    3049#   3051    3087#   3089    3140#   3142    3173#   3175    3232#   3234    3328#
                    3330    3406#   3408    3541#   3543    3818#   3820    4080#   4082    4215#   4217    4358#   4360
                    4489#   4491    4619#   4621    4690#   4705#   4707    4918#   4937#   4939    5048#   5050    5169#
                    5171    5293#   5295    5387#   5389    5514#   5516    5640#   5642    5766#   5768    5861#   5863
                    5976#   5978    6107#   6109    6241#   6243    6375#   6377    6489#   6491    6589#   6591    6757#
                    6759    6900#   6902    7000#   7002    7103#   7118#   7120    7184#   7199#   7201    7265#   7280#
                    7282    7342#   7357#   7359    7418#   7420    7486#   7488    7560#   7578#   7580    7692#   7708#
                    7710    7818#   7820    8003#   8018#   8020    8121#   8123    8231#   8233    8341#   8343
$OCNT   056676      11426*  11455*  11468#
$OMODE  056700      11421*  11425*  11430   11433*  11444*  11470#
$OVER   054170      10829   10845   10853   10863   10871#
$PASS   001100        242#   4114    8371*   8373    8406*   8407*   8415    8428   10859   10875
$POWER  057144      11552   11559#
$PWRAD  057132      11554#
$PWRDN  056772       1188   11521#  11549
$PWRMG  057126      11552#
$PWRUP  057044      11531   11537#
$QUES   001222        288#  11012   11172   11216   11232   11290   11293   11338
$RDCHR  055522      11185#  11511
$RDDEC= ****** U    11514
$RDLIN  055612      11208#  11512
$RDOCT  055772      11254#  11513
$RDSZ = 000011      11201#
$REGAD  001160        271#
$REGO   001162        273#
$REG1   001164        274#
$REG2   001166        275#
$REG3   001170        276#
$REG4   001172        277#
$REG5   001174        278#
$RTNAD  041334       8427#
$R2A  = ****** U    11514
$SAVRE= ****** U    11514
$SAVR6  057142      11530*  11538   11539*  11540*  11558#
$SCOPE  053714       1182   10815#
$SETUP= 000117       1173#   1181    1182    1184    1186    1188    1190    1191    1193    1260    8404   10816   11055
                    11060   11061   11091   11239   11308   11329   11337
$SS1  = 000000       1212#
$STUP = 177777       1173#
$SVLAD  054142      10837   10866#
$SVPC = 000200        176#    181
$SWR  = 167770          1#      11      41      42      43      44      45      46      47      48     285     286     287
                     1190    1191    1193    1194    1295    1345    1374    1393    1556    1679    1711    1747    1769
                     1823    1847    1871    1894    1952    1978    2004    2029    2050    2086    2262    2511    2579
                     2676    2742    2808    2882    2924    3009    3056    3094    3147    3180    3240    3338    3415
                     3552    3828    4089    4225    4368    4499    4629    4694    4717    4922    4948    5059    5179
                     5303    5396    5523    5649    5776    5871    5986    6118    6256    6386    6500    6602    6767
                     6910    7019    7107    7132    7188    7213    7269    7294    7346    7371    7432    7500    7564
                     7589    7696    7719    7834    8007    8029    8135    8245    8351    8399    8405    8420    8426
                     8428   10807   10808   10809   10810   10811   10828   10840   10842   10843   10846   10847   10848
                    10855   10856   10857   10868   10871   10874   11299   11300   11301   11302   11303   11314   11321
                    11326   11330   11338   11555
$SWRMK= 000000         48      49   10811   10812   10844
$TIMES  001212        285#   1190*   1295*   1345*   1374*   1393*   1556*   8351*   8405*  10855*  10862   10865*  10874
$TKB    001146        264#  11015   11036   11047   11072   11100   11127
```

KB

```
$TKCNT  054652          11016# 11031* 11061  11078* 11192  11194*
$TKINT  054672           1216   1397   2607   8564   9670  11031# 11052  11113
$TKQEN= 054671          11020# 11086  11197
$TKQIN  054654          11017# 11032* 11033  11084* 11085* 11086  11088*
$TKQOU  054656          11018# 11033* 11195  11196* 11197  11199*
$TKQSR  054660          11019# 11032  11088  11199
$TKS    001144            263# 11015  11037* 11068* 11070  11076* 11098  11114* 11124  11136* 11156*
$TKSRV  054742          11034  11047#
$TMP0   001176            279# 1474*  1475*  8757*  8762   8767   8771   8881*  8895   8994*  9040*  12307
$TMP1   001200            280# 1310*  4743*  4768   4835*  4839   4853*  4883   6280*  6335*  8758*  8762   8767
                         8882* 8897   12305  12307  12320
$TMP2   001202            281# 4771*  4772*  4841*  4842*  8761*  8764*  8766*  8769*  8996*  9038*  12320
$TMP3   001204            282# 4775*  4845*  8755*  8756*  8995*  8997   8998*  12307  12320
$TMP4   001206            293# 4765*  4774   4796*  4825*  4844   8997*  9021*
$TMP5   001210            284# 3849*  3851*  3852   3858*  3860*  3861   3930   8993*  8999*  9005   9118*  9197
                         9219
$TN    = 000124             1#   11   1275   1288   1295#  1297   1337   1345#  1347   1353   1370   1374#  1376
                         1382 1389   1393#  1396   1538   1543   1556#  1559   1670   1679#  1682   1704   1711#
                         1717 1724   1740   1747#  1749   1762   1769#  1771   1816   1823#  1825   1840   1847#
                         1849 1864   1871#  1873   1887   1894#  1895   1945   1952#  1955   1971   1978#  1981
                         1997 2004#  2006   2022   2029#  2031   2043   2050#  2052   2077   2086#  2087   2250
                         2262# 2489  2501   2511#  2512   2570   2579#  2580   2665   2676#  2677   2683   2722
                         2727 2742#  2743   2749   2789   2799   2808#  2814   2875   2882#  2888   2908   2915
                         2924# 2925  2931   2990   3002   3009#  3010   3016   3038   3045   3049   3056#  3057
                         3078 3087   3094#  3095   3101   3132   3140   3147#  3148   3166   3173   3180#  3182
                         3199 3232   3240#  3242   3328   3338#  3339   3406   3415#  3541   3552#  3553   3799
                         3818 3828#  3829   4080   4089#  4090   4210   4215   4225#  4226   4358   4368#  4369
                         4489 4499#  4500   4619   4629#  4630   4690   4694#  4695   4705   4717#  4718   4918
                         4922# 4923  4937   4948#  4949   5014   5037   5048   5059#  5060   5124   5152   5169
                         5179# 5180  5248   5275   5293   5303#  5304   5387   5396#  5397   5476   5501   5514
                         5523# 5524  5603   5628   5640   5649#  5650   5729   5754   5766   5776#  5777   5781
                         5832 5853   5861   5871#  5873   5926   5967   5976   5986#  5993   6095   6107   6118#
                         6126 6221   6241   6256#  6258   6365   6375   6386#  6388   6480   6489   6500#  6501
                         6507 6589   6602#  6610   6741   6757   6767#  6771   6849   6888   6900   6910#  6915
                         6991 7000   7019#  7021   7103   7107#  7108   7118   7132#  7135   7184   7188#  7189
                         7199 7213#  7216   7265   7269#  7270   7280   7294#  7297   7342   7346#  7347   7357
                         7371# 7374  7418   7432#  7435   7486   7500#  7503   7560   7564#  7565   7578   7589#
                         7592 7692   7696#  7697   7708   7719#  7722   7818   7834#  7835   8003   8007#  8008
                         8018 8029#  8030   8121   8135#  8136   8231   8245#  8246   8341   8351#
$TPB    001152            266# 11001* 11012
$TPFLG  001157            270# 10959  11012
$TPS    001150            265# 10999  11012
$TRAP   056702           1186  11479#
$TRAP2  056724          11490# 11501
$TRP   = 000016         11494# 11503# 11504# 11505# 11506# 11507# 11508  11509# 11510  11511# 11512# 11513# 11514#
                        11515# 11516# 11517#
$TRPAD  056736          11484  11501#
$TSTNM  001102            243#  8366*  8404*  10806  10844  10866*  10871  10875  11313  11338
$TTYIN  055720          11209  11210  11227  11231#
$TYPBN= ****** U        11507
$TYPDS  054206          10887# 11506
$TYPE   054432          10959# 11494  11502
$TYPEC  054602          10980  10987  10994  10999#  11000  11158
$TYPEX  054650          11005  11007  11010#
$TYPOC  056500          11424# 11503
$TYPON  056514          11423  11426# 11505
```

```
$TYPOS  056454          11419#  11504
$XTSTR  053746          10831#
$$GET4= 000000           8420#
$OFILL  056677          11420*  11424*  11434   11469#
$40CAT= ****** U        10828   11323
.     = 067750            164#    168#    176     177#    179#    181#    185#    187#    232#    240#    291     1074#   1147#
                         1148#   1149#   1179    1193    1194    1226#   1231#   1244#   1248#   1318#   1404#   1444#   1448#
                         1456#   1506#   1513#   1519#   1607#   1648#   1654#   1660#   1686#   1690#   1700#   4122#   8362#
                         8428    8432    8476#   8491#   9681#  10307#  10308#  10309#  10310#  10311#  10312#  10314#  10315#
                        10316#  10317#  10318#  10874   10875   10941#  11012   11015   11019#  11020   11021#  11231#  11232
                        11239   11293   11338   11393#  11533   11557   12290#  12420#
```

```
CHECK1    36#  5015  5125  5249  5477  5604  5730  6850
CHECKP    36#  3810  4795  4985  5098  5220  5339  5446  5573  5699  5816  5910  6260  6425  6544
          6819  6947  7023  7627  7757  7888  8056  8161  8266  8847  8947  9114
COMMEN     12   161#  2217
ENDCOM     19   161#  2242
ERROR      55#  1311  1362  1385  1731  1756  1789  1800  1809  1832  1856  1880  1909  1931  1962
           1988  2013  2038  2059  2174  2293  2494  2561  2655  2695  2704  2716  2724  2760  2775
           2786  2793  2836  2853  2855  2904  2910  2942  2962  2977  2985  2996  3031  3041  3071
           3081  3119  3134  3161  3169  3194  3202  3268  3279  3294  3318  3398  3466  3534  3803
           3910  3963  4024  4074  4156  4201  4303  4349  4433  4479  4565  4610  4681  4776  4846
           4913  5010  5038  5040  5120  5153  5155  5242  5276  5278  5358  5467  5502  5504  5594
           5629  5631  5720  5755  5757  5845  5848  5950  5952  6050  6056  6096  6098  6181  6186
           6223  6225  6292  6301  6314  6321  6349  6359  6366  6463  6465  6560  6570  6578  6671
           6676  6686  6693  6700  6707  6743  6745  6840  6875  6877  6889  6973  6979  6992  7092
           7656  7659  7676  7786  7789  7806  7968  7989  7991  8112  8221  8334  8672  8678  8682
           8694  8699  8702  8721  8728  8734  8739  8772  9162  9164  9188  9190  9205  9216  9226
           9236  9307  9513  9519  9587
ESCAPE     161#
GETPRI     161#
GETSWP      1#   161#  1260
HCOMPR     36#  7133  7214  7433
HCOMPW     36#  7295  7372  7501
MAKECL     36#  4690  4918  7103  7184  7265  7342  7560  7691  8003
MSG      1287#  1290  1336#  1339  1542#  1545  1670#  1672  1704#  1706  1739#  1742  1761#  1764  1815#
         1818  1839#  1842  1863#  1866  1886#  1889  1944#  1947  1970#  1973  1996#  1999  2021#  2024
         2042#  2045  2076#  2079  2248#  2252  2500#  2503  2569#  2572  2664#  2667  2726#  2729  2798#
         2801  2874#  2877  2914#  2917  3001#  3004  3048#  3051  3086#  3089  3139#  3142  3172#  3175
         3231#  3234  3327#  3330  3405#  3408  3540#  3543  3817#  3820  4079#  4082  4214#  4217  4357#
         4360  4488#  4491  4618#  4621  4704#  4707  4936#  4939  5047#  5050  5168#  5171  5292#  5295
         5386#  5389  5513#  5516  5639#  5642  5765#  5768  5860#  5863  5973#  5978  6106#  6109  6239#
         6243  6374#  6377  6488#  6491  6588#  6591  6756#  6759  6899#  6902  6999#  7002  7577#  7580
         7707#  7710  7816#  7820  8017#  8020  8120#  8123  8230#  8233  8341#  8343
MSGA     7118#  7120  7201  7420
MSGB     7279#  7282  7359  7488
MULT      161#
NEWTST    161#  1288  1337  1370  1389  1543  1670  1704  1740  1762  1816  1840  1864  1887  1945
          1971  1997  2022  2043  2077  2250  2501  2570  2665  2727  2799  2875  2915  3002  3049
          3087  3140  3173  3232  3328  3406  3541  3818  4080  4215  4358  4489  4619  4690  4705
          4918  4937  5048  5169  5293  5387  5514  5640  5766  5861  5976  6107  6241  6375  6489
          6589  6757  6900  7000  7103  7118  7184  7199  7265  7280  7342  7357  7418  7486  7560
          7578  7692  7708  7817  8003  8018  8121  8231  8341
POP       161#  8524  8621  8809  8905  9043  9091  9269  9309  9538  9602  9642  9912  10066  10136
         10204  10278  10471  10629  10744  10784  10923  11282  11542  11543
PUSH      161#  8514  8608  8799  8872  8985  8988  9063  9250  9287  9398  9562  9616  9824  9971
         10090  10158  10234  10353  10509  10675  10765  10887  11256  11523  11529
REPORT    161#
RFORGC     36#  5037  5152  5275  5501  5628  5754  6095  6221  6741
RH70CK     36#  1348  1712  2678  2744  2809  2883  2926  3011  3096  6502
SAVE       36#  11309
SAVTST     36#  1297  1347  1376  1396  1559  1679  1711  1724  1749  1771  1825  1849  1873  1895
          1955  1981  2006  2031  2052  2087  2512  2580  2677  2743  2808  2882  2925  3010  3057
          3095  3148  3182  3242  3339  3415  3553  3829  4090  4226  4369  4500  4630  4695  4718
          4923  4949  5060  5180  5304  5397  5524  5650  5777  5781  5873  5993  6126  6258  6388
          6501  6610  6771  6915  7021  7108  7135  7189  7216  7270  7297  7347  7374  7435  7503
          7565  7592  7697  7722  7835  8008  8030  8136  8246
SCOPE      56#  1294  1344  1373  1392  1555  1678  1710  1746  1768  1822  1846  1870  1893  1951
```

```
        1977    2003    2028    2049    2085    2261    2510    2578    2675    2741    2807    2881    2923    3008    3055
        3093    3146    3179    3239    3337    3414    3551    3827    4088    4224    4367    4498    4628    4693    4716
        4921    4947    5058    5178    5302    5395    5522    5648    5775    5870    5985    6117    6255    6385    6499
        6601    6766    6909    7018    7106    7131    7187    7212    7268    7293    7345    7370    7431    7499    7563
        7588    7695    7718    7833    8006    8028    8134    8244    8350    8403
SETPRI   161#  11188
SETTRA 11494#  11503   11504   11505   11506   11508   11510   11511   11512   11513   11514   11515   11516
SETUP    161#   1174
SKIP      36#    161#   1275    1382    1538    1682    2489    2722    2789    2908    2990    3038    3078    3132    3166
        3199    3799    5014    5124    5248    5476    5603    5729    5832    5853    5926    5967    6365    6480    6849
        6888    6991
SLASH    161#
SMORE     36#  10817
SPACE    161#
STARS      27      34    161#    174     231     236     291     735     737     775     777    1288    1293    1337    1343
        1370    1372    1389    1391    1486    1496    1543    1554    1589    1600    1623    1661    1670    1677    1704
        1709    1740    1745    1762    1767    1816    1821    1840    1845    1864    1869    1887    1892    1945    1950
        1971    1976    1997    2002    2022    2027    2043    2048    2067    2073    2077    2084    2135    2145    2229
        2232    2250    2260    2501    2509    2570    2577    2665    2674    2727    2740    2799    2806    2875    2880
        2915    2922    3002    3007    3049    3054    3087    3092    3140    3145    3173    3178    3207    3217    3224
        3228    3232    3238    3328    3336    3406    3413    3541    3550    3818    3826    4080    4087    4215    4223
        4358    4366    4489    4497    4619    4627    4690    4692    4705    4715    4918    4920    4937    4946    5048
        5057    5169    5177    5293    5301    5362    5382    5387    5394    5514    5521    5640    5647    5766    5774
        5861    5869    5976    5984    6107    6116    6241    6254    6375    6384    6489    6498    6589    6600    6757
        6765    6900    6908    7000    7017    7103    7105    7118    7130    7184    7186    7199    7211    7265    7267
        7280    7292    7342    7344    7357    7369    7418    7430    7486    7498    7550    7556    7560    7562    7578
        7587    7692    7694    7708    7717    7818    7832    8003    8005    8018    8027    8121    8133    8231    8243
        8341    8349    8396    8434    8453    9734    9735    9794    9797    9921    9943    9950    9952   10077   10079
       10144   10146   10215   10218   10293   10299   10325   10327   10481   10483   10641   10643   10754   10756   10803
       10877   10944   11014   11091   11106   11177   11201   11242   11295   11340   11396   11473   11519   11535
SWRSU    161#   1195#
TJUMP     36#   1353    1717    2683    2749    2814    2888    2931    3016    3045    3101    4210    6507
TRMTRP 11494#
TSCLR     36#   2303    2313    2336    2346    2356    2366    2376    2386    2396    2415    2459    2470
TSCLR1    36#   2436    2448
TSCLR2    36#   3598    3609    3620    3660    3670    3690    3690    3700    3710    3729    3770    3780
TSCLR3    36#   3749    3760
TSCLR4    36#   2323
TSCLR5    36#   3632
TYPBIN   161#
TYPDEC   161#   8415   11380
TYPNAM   161#
TYPNUM   161#
TYPOCS   161#   2189    2202
TYPOCT   161#   1614    1620   11119   11352   11377
TYPTXT   161#   1222    1228    1233    1237    1241    1245    1249    1253    1276    1315    1319    1401    1405    1409
        1413    1440    1445    1449    1453    1503    1510    1516    1604    1610    1616    1644    1650    1656    1683
        1687    1691    1697    4119    8353    8359    8460    8466    8473    8477    8481    8488    8492    9659    9666
        9678    9684    9690    9694    9698    9704    9710   10791
WDATAR    36#   5779    7590    7720
XREAD     36#   5399    5526    5652    6772
$$CMRE   234#    273     274     275     276     277     278
$$CMTM   234#    279     280     281     282     283     284
$$ESCA   161#
$$NEWT   161#   1288    1337    1370    1389    1543    1670    1704    1740    1762    1816    1840    1864    1887    1945
        1971    1997    2022    2043    2077    2250    2501    2570    2665    2727    2799    2875    2915    3002    3049
```

```
        3087    3140    3173    3232    3328    3406    3541    3818    4080    4215    4358    4489    4619    4690    4705
        4918    4937    5048    5169    5293    5387    5514    5640    5766    5861    5976    6107    6241    6375    6489
        6589    6757    6900    7000    7103    7118    7184    7199    7265    7280    7342    7357    7418    7486    7560
        7578    7692    7708    7818    8003    8018    8121    8231    8341
$$SET  11494#  11503   11504   11505   11506   11508   11510   11511   11512   11513   11514   11515   11516
$$SKIP   161#
.EQUAT     1#     51
.FLOAT    36#   1722    1747    1823    1847    1871    1953    1979    2004    2029    2050
.HEADE     1#
.KT11      1#    196
.SETUP     1#   1172
.SWRHI     1#     37
.SWRLO     1#     49#     50
.$ACT1     1#    172
.$CATC     1#    162
.$CMTA     1#    234
.$EOP      1#   8394
.$ERRO     1#  11293
.$ERRT     1#  11338
.$POWE     1#  11517
.$RDOC     1#  11240
.$READ     1#  11012
.$SCOP     1#  10801
.$TRAP     1#  11471
.$TYPD     1#  10875
.$TYPE     1#  10942
.$TYPO     1#  11394

. ABS.  067750     000

 ERRORS DETECTED:  0

 ,DZRJGA/SOL/CRF=DZRJGA
 RUN-TIME: 34 33 3 SECONDS
 RUN-TIME RATIO: 2203/71=30.7
 CORE USED:  29K  (57 PAGES)
```