RP04/5/6

RP04/5/6 CTRLR 1

CZRJID0

AH 9220D MC

FICHE 1 OF 2

NOV 1979

digital

MADE IN USA

RP04/5/6

RP04/5/6 CTRLR 1

CZRJIDO

AH 9220D MC

FICHE 2 OF 2

NOV 1979

.REM @

IDENTIFICATION

PRODUCT CODE: AC-9218D-MC

PRODUCT NAME: CZRJIDO RP04/5/6 FUNCTIONAL CONTROLLER TEST PART I

DATE CREATED: MAY, 1979

MAINTAINER: DIAGNOSTIC ENGINEERING

AUTHOR: PETE BLACKSTONE

CONTENTS

1.0 ABSTRACT

THIS DIAGNOSTIC IS USED TO TEST RP04/5/6 DEVICE CONTROL LOGIC CONNECTED TO AN RH11 OR RH70 CONTROLLER.

IT USES THE DISK SURFACE AND THE DRIVE MECHANICS TO PROVE THE PROPER WORKING OF THE SUBSYSTEM. IT DOES NOT NEED A FORMATTED DISK PACK. A DISK PACK WITH NO VITAL INFORMATION WRITTEN ON IT IS ESSENTIAL. AFTER A SUCCESSFUL RUN (WITH NO ERRORS) OF THIS DIAGNOSTIC IT CAN BE ASSERTED THAT THE DCL IN THE RP04/5/6 SUBSYSTEM WORKS SUCCESSFULLY WHILE STANDING ALONE. SYSTEMS INTERACTION AND DRIVE TIMING IS LEFT TO OTHER DIAGNOSTICS. THIS IS WITH THE ASSUMPTION THAT PROGRAMS DZRJGA AND DZRJHA HAVE BEEN RUN SUCCESSFULLY.

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 CONTROLLER, A DISK CONTROL LOGIC (DCL), A DEC 733 DISK DRIVE, AND ITS APPROPRIATE DISK PACK. THE DISK PACK NEED NOT BE FORMATTED. USED SECTION OF THE DISK SURFACE SHALL BE GOOD (HOLE FREE). THE SURFACE FOR THE FOLLOWING SECTORS MUST BE GOOD, THAT IS, FREE OF ANY HOLES OR SURFACE IRREGULARITY BEFORE ANY DATA ERROR CAN BE ATTRIBUTED TO THE LOGIC.

CYLINDER 00, TRACK 00, SECTOR 00
CYLINDER 00, TRACK 00, SECTOR 01
CYLINDER 00, TRACK 18, SECTOR 21
CYLINDER 01, TRACK 00, SECTOR 00
CYLINDER 02, TRACK 00, SECTOR 00
CYLINDER 03, TRACK 00, SECTOR 00
CYLINDER 04, TRACK 00, SECTOR 00
CYLINDER 05, TRACK 00, SECTOR 00
CYLINDER 05, TRACK 07, SECTOR 04
CYLINDER 06, TRACK 00, SECTOR 00
CYLINDER 07, TRACK 00, SECTOR 00
CYLINDER 08, TRACK 00, SECTOR 00
CYLINDER 09, TRACK 18, SECTOR 21
CYLINDER 410,TRACK 18, SECTOR 21

2.2 STORAGE

THIS PROGRAM REQUIRES 16K WORDS OF MEMORY

2.3 PRELIMINARY PROGRAMS

THIS PROGRAM ASSUMES THAT MAINDEC-11-DZRJG-(LATEST REV) HAS BEEN RUN WITHOUT ERRORS.
AND IT ASSUMES THAT MAINDEC-11-DZRJH-(LATEST REV) HAS BEEN RUN WITHOUT ERRORS.

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 FRONT PANEL, OR IN THE SOFTWARE SWITCH REGISTER IF THE OPERATOR SO DESIRES. SEE PARAGRAPH 5.1 FOR A DESCRIPTION OF SOFTWARE SWITCH REGISTER OPERATION.

4.1 CONTROL SWITCH SETTINGS

SEE SECTION 5.1

4.2 STARTING ADDRESS

START AT ADDRESS 200---FOR NORMAL RUN
START AT ADDRESS 204---TO SELECT NON-DEFAULT ADDRESSES
START AT ADDRESS 210---FOR UNIT SELECTION
START AT ADDRESS 220---FOR NO MANUAL INTERVENTION

200 START
ALL SWITCHES MUST BE DOWN FOR WORST CASE RUN. WITH THIS STARTING ADDRESS ALL THE RP04/5/6S ON THE SYSTEM WILL BE TESTED ONE AT A TIME BEFORE "END PASS" IS PRINTED OUT. TESTING WILL START WITH THE LOWEST UNIT NUMBER DRIVE THAT IS POWERED UP (THAT IS THE LOWEST UNIT NUMBER RHAS REGISTER THAT RESPONDS)THEN GO ON TO THE NEXT HIGHER UNIT NUMBER THAT IS POWERED UP.

204 RESTART
SAME AS START 200 WITH THE FOLLOWING EXCEPTION: THE PROGRAM WILL INTERROGATE THE OPERATOR FOR A NON-STANDARD C.S.R AND VECTOR ADDRESS BEFORE STARTING. ONCE THE QUESTIONS HAVE BEEN CORRECTLY ANSWERED, AND IT IS ALSO NECESSARY TO SELECT A PARTICULAR UNIT FOR TEST (TYPICAL PROGRAM EXECUTION FROM ADDRESS 210), OR IT IS NECESSARY TO RUN THE PROGRAM WITHOUT MANUAL INTERVENTION (TYPICAL PROGRAM EXECUTION FROM ADDRESS 220), THE PROCESSOR MAY BE HALTED AND RESTARTED FROM THE DESIRED RESTART ADDRESS. IF ALL UNITS ARE TO BE CHECKED, THE PROCESSOR NEED NOT BE TOUCHED. THE PROGRAM WILL AUTOMATICALLY RESTART AT ADDRESS 200 AFTER RECEIVING THE NEW DEVICE PARAMETERS.

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

220 START
ALL SWITCHES MUST BE DOWN FOR WORST CASE RUN. WITH THIS STARTING ADDRESS THE PROGRAM WILL NOT RUN THOSE TESTS THAT NEED

MANUAL INTERVENTION. THIS IS RECOMMENDED ONLY FOR DEBUGGING WHERE THE ERROR IS NOT IN A TEST THAT REQUIRES MANUAL INTERVENTION

4.3 PROGRAM AND/OR OPERATOR ACTION

1. LOAD THE PROGRAM INTO MEMORY.
2. SET STARTING ADDRESS ON THE SWITCH REGISTER
3. PRESS "LOAD ADDRESS".
4. SET "OPERATIONAL SWITCH SETTINGS" (SEE SECTION 5.1) WORST CASE IS ALL SWITCHES DOWN.
5. PRESS "START".
6. FOR THE FIRST PASS EACH TEST WILL BE EXECUTED ONCE ON THE DRIVES PRESENT OR DRIVE SELECTED BEFORE "END PASS" IS PRINTED. THE FIRST PASS WILL REQUIRE OPERATOR INTERVENTION IF THE PROGRAM IS NOT RUN UNDER AN "ACT-11" MONITOR. THE SECOND AND SUBSEQUENT PASSES WILL EXECUTE EACH TEST FOUR TIMES ON EACH DRIVES PRESENT OR DRIVE SELECTED BEFORE "END PASS" IS PRINTED. THE SECOND AND SUBSEQUENT PASSED DO NOT NEED ANY OPERATOR INTERVENTION.

5.0 OPERATING PROCEDURE

5.1 OPERATIONAL SWITCH SETTINGS

IF THE PROGRAM IS BEIDNG RUN ON A SWITCHLES PROCESSOR (I. E. AN 11/34) IT WILL DETERMINE THAT A HARWARE SWITCH REGISTER IS NOT PRESENT, AND WILL USE "SOFTWARE" SWITCH REGISTER. THE SETTINGS OF THE SWITCHES ARE CONTROLLED THROUGH A KEYBOARD ROUTINE WHICH IS CALED BY TYPING A 'COBNTROL G'. THE PROGRAM WILL RECOGNIZE A 'CONTROL G' AT ANY TIME EXCEPT WHEN IT IS AR A HIGHER PRIORITY PROCESSING AN RP04/5/6 INTERRUPT. THE "SOFTWARE" SWITCH VALUE S ARE ENTERED AS AN OCTAL NUMBER IN RESPONSE TO PROMPTING 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 REGISTER, 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 CONTROLLER SELECT
THIS SWITCH 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: 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: 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 - TYPE ALL REGISTERS WITH ERROR 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. THAT IS ON FINDING AN ERROR INSTEAD OF ONLY GIVING THE ERROR MESSAGE AND RELEVANT REGISTERS AS WILL BE DONE IF SWITCH 11 IS NOT SET BUT WILL ALSO GIVE ALL THE REGISTER CONTENTS (EXCEPT "DATA BUFFER" RHDB).

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 STATFD ADDRESS TO DETERMINE THE CAUSE OF THE ERROR.

6.1

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 OPERATIOR. 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 FILURE CONTINUES TO OCCUR, THERE ARE TWO OPTIONS OPEN TO THE OPERATOR:

1. 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.

2. GO BACK AND RERUN DZRPS, AS IT IS QUITE POSSIBLE THAT A HARD FAILURE HAS OCCURRED IN ONE OF THE HARDWARE REGISTERS.

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

7.0 RESTRICTIONS

BEFORE STARTING THE PROGRAM THE OPERATOR MUST HAVE THE DRIVE PORT SWITCH LOCKED EITHER ON PORT A OR PORT B BUT MUST NEVER LEAVE IT IN THE PROGRAMMABLE STATE.

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 ON AN RH70. THIS IS BECAUSE THE ROUTINE WHICH GETS "SOFTWARE" SWITCH SETTINGS IS NOT OPERABLE WHEN IN CHAIN MODE.

8. MISCELLANEOUS

8.1 EXECUTION TIME

THE FIRST PASS OF THE PROGRAM WILL TAKE APPROXIMATELY 1 MINUTES PROVIDED AN OPERATOR IS PRESENT TO CARRY OUT THE TYPED INSTRUCTIONS IMMEDIATELY. SUBSEQUENT PASSES WILL TAKE 30 SECONDS WHETHER AN OPERATOR IS THERE OR NOT.

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 THE 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

9.1 LOGIC DIVISION IN HARDWARE MODULES

REGISTER BOARD (RG) - ERROR REGISTER 1 STATUS REGISTERS
MUX FOR REGISTERS GO HANDLING REGISTER
DECODE COMMAND DECODE EXECUTION OF
MECH. COMMANDS

SYNC. DATA BOARD (SN) - DATA CONTROL PARALLEL TO SERIAL
SYNC. BYTE DETECT.

SEEK AND SEARCH (SS) - SEEK LOGIC SEARCH LOGIC HEADER
HANDLING.

ERROR CORRECTION (EC) - ECC LOGIC ERROR REGISTER 2 & 3
MUX FOR ERROR REG. 2 & 3 LOOK AHEAD
REG. SECTOR COUNTER DATA FORMATION
RING COUNTER.

DUAL PORT (DP) - DUAL PORT ARBITRATION ATTENTION LOGIC
SERIAL NO REGISTER MASS BUS REGISTER
STORAGE

9.2 DISK SURFACE USAGE

SYMBOLS USED
C = CYLINDER
T = TRACK
S = SECTOR
W = WRITE
R = READ
TT = TEST NUMBER

C0, T0, S0
TT22-W,R, TT23-R, TT24-W,R, TT25-W,R, TT26-W,R, TT35-W,R, TT37-W, TT50-W, TT51-W,R, TT52-W,R, TT55-W,R

C0, T0, S1
TT27-W,R, TT37-W,R, TT40-R, TT41-W,R, TT42-W,R, TT43-W,R

C0, T18, S21
TT30-W, TT31-W,R

C1, T0, S0
TT30-W,R, TT31-W,R, TT53-W,R, TT54-W,R

C1, T18, S21
TT31-W

C2, T0, S0
TT31-W,R

C2, T18, S21
TT31-W

C3, T0, S0

```
                    TT31-W,R

                    C3, T18, S21
                    TT31-W

                    C4, T0, S0
                    TT31-W,R

                    C4, T18, S21
                    TT31-W

                    C5, T0, S0
                    TT31-W,R

                    C5, T7, S4
                    TT33-W,R, TT34-W,R

                    C5, T18, S21
                    TT31-W

                    C6, T0, S0
                    TT31-W,R

                    C6, T18, S21
                    TT31-W

                    C7, T0, S0
                    TT31-W,R

                    C7, T18, S18
                    TT31-W

                    C8, T0, S0
                    TT31-W,R

                    C8, T18, S21
                    TT31-W

                    C9, T0, S0
                    TT31-W

                    C9, T18, S21
                    TT31-W, TT32-R

                    C10, T0, S0
                    TT31-W,R

                    C410, T18, S21
                    TT36-W,R, TT50-W,R

9.3     THE FOLLOWING SECTION DESCRIBES EACH TEST AND SUBROUTINES
        IN DETAIL AND CAN 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  CZRJIDO, RP04/5/6 FCTNL CTLR1
                                    ;*COPYRIGHT (C) 1976,1977
                                    ;*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:         ALL MACRO CALLS BEGINNING WITH ".$" ARE SUPPLIED FROM AN
                                    ;*       EXTERNAL SYSMAC.SML 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           TYPE ALL REG. WITH ERROR IF SW8 LOW
                        ;*          5           MULT ADDR PLUG TEST IF SW08 IS LOW
```

```
                        .SBTTL  BASIC DEFINITIONS

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

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

                        ;*GENERAL PURPOSE REGISTER DEFINITIONS
000000                  R0=     %0              ;;GENERAL REGISTER
000001                  R1=     %1              ;;GENERAL REGISTER
000002                  R2=     %2              ;;GENERAL REGISTER
000003                  R3=     %3              ;;GENERAL REGISTER
000004                  R4=     %4              ;;GENERAL REGISTER
000005                  R5=     %5              ;;GENERAL REGISTER
000006                  R6=     %6              ;;GENERAL REGISTER
000007                  R7=     %7              ;;GENERAL REGISTER
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  041140                  $ENDAD                  ;;1)SET LOC.46 TO ADDRESS OF $ENDAD IN .$EOP
                        000052          .=52
000052  002000                  .WORD   2000            ;;2)SET LOC.52 TO 2000
                        000200          .=$SVPC                 ;; RESTORE PC

                                .SBTTL  STARTING ADDRESSES

                        000200          .=200
000200  000137  005012  RA:     JMP     @#BEGIN         ;NORMAL START
000204  000137  043660  ADDMOD: JMP     @#BASECH        ;START FOR ADDRESS-MODIFICATION
000210  000137  004776          JMP     @#BEGIN2        ;JUMP TO SELECT DRIVE START
                        000220          .=220
000220  000137  004762          JMP     @#BEGIN1        ;JUMP TO NO OPERATOR TESTS START


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

                        ;*STARTING ADDRESS 204 FOR NON-DEFAULT ADDRESS PARAMETERS
                        ;*AUTO RESTART AT ADDRESS 200 AFTER LOADING PARAMETERS

                        ;*STARTING ADDRESS 210 WILL TEST ONLY ONE SPECIFIED DRIVE

                        ;*STARTING ADDRESS 220 WILL JUMP OVER THE TESTS REQUIRING AN OPERATOR
                        ;*AT THE 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
```

```
                        .SBTTL  COMMON TAGS

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

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

```
   911  001224  000012                  $LF:    .ASCIZ  <12>            ;;LINE FEED
   912                                  ;;**********************************************************************
```

```
                                .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  051166                          EM1             ;RP04 DID NOT INTERRUPT
                                                        ;WAITED ON BIT DID NOT OCCUR
001230  067273                          DH1             ;PC
                                                        ;WAT PC
                                                        ;BIT WAITED
                                                        ;REG ADDRESS
                                                        ;REG CONTENTS
                                                        ;RHCS1 CONTENTS
001232  071620                              DT1         ;$ERRPC,WAITPC,WAITBT,WAITRE,$BDDAT,CS1
001234  072126                              DF1         ;0,0,0,0,0,0

                                ;*ITEM2
001236  051215                          EM2             ;INTERRUPT ENABLE BIT DOWN BUT
                                                        ;WAITED ON BIT DID NOT OCCUR
001240  067273                          DH1             ;PC
                                                        ;WAT PC
                                                        ;BIT WAITED
                                                        ;REG ADDRESS
                                                        ;REG CONTENTS
                                                        ;RHCS1 CONTENTS
001242  071620                              DT1         ;$ERRPC,WAITPC,WAITBT,WAITRE,$BDDAT,CS1
001244  072126                              DF1         ;0,0,0,0,0,0

                                ;*ITEM3
001246  051304                          EM3             ;RP04 DID NOT INTERRUPT WHEN
                                                        ;WAITED ON BIT DID SET
001250  067273                          DH1             ;PC
                                                        ;WAT PC
                                                        ;BIT WAITED
                                                        ;REG ADDRESS
                                                        ;RHCS1 CONTENTS
001252  071620                              DT1         ;$ERRPC,WAITPC,WAITBT,WAITRE,$BDDAT,CS1
001254  072126                              DF1         ;0,0,0,0,0,0

                                ;*ITEM4
```

```
                                        EM4             ;WAITED ON BIT DID SET BUT
                                                        ;TIME IS IN ERROR
                                                        ;TIME IS GIVEN IN 10 MICRO SEC.
                                                        ;(DECIMAL)
                                        DH4             ;PC
                                                        ;WAT PC
                                                        ;BIT WAITED
                                                        ;REG ADDRESS
                                                        ;TIME IN 10 MSEC
                                        DT4             ;$ERRPC,WAITPC,WAITBT,WAITRE,$BDDAT,WAITIM
                                        DF4             ;0,0,0,0,0,1

                                ;*ITEM5
                                        EM5             ;RHAS DOES NOT CLEAR BY
                                                        ;MOVING IN ALL ONES
                                        DH5             ;PC
                                                        ;REG. ADDR.
                                                        ;GOOD DATA
                                                        ;RECEIVED DATA
                                        DT5             ;$ERRPC,REGADR,$GDDAT,$BDDAT
                                        DF5             ;0,0,0,0

                                ;*ITEM6
                                        EM6             ;LOADING RHER1 FOR ALL
                                                        ;UNITS DID NOT SET ANY BITS
                                                        ;IN RHAS-NO UNITS PRESENT
                                        DH6             ;PC
                                                        ;REG ADDR
                                                        ;RECEIVED DATA
                                        DT6             ;$ERRPC,REGADR,$BDDAT
                                        DF6             ;0,0,0

                                ;*ITEM7
                                        EM7             ;SPECIFIED REGISTER NONEXISTANT
                                                        ;SO ABORT PROGRAM
                                        DH7             ;PC
                                                        ;ADDR. OF REG.
                                        DT7             ;$ERRPC,TEMP1
                                        DF7             ;0,0

                                ;*ITEM10
                                        EM10            ;STOPED DRIVE HAS MOL BIT
                                                        ;IN RHDS1 = 1
                                        DH10            ;PC
                                                        ;TEST NO
                                                        ;FAILING REG ADDR
                                                        ;CONTENTS OF RHCS1
                                                        ;CONTENTS OF RHCS2
                                                        ;CONTENTS OF RHDS1
                                                        ;CONTENTS OF RHER1
                                        DT10            ;$ERRPC,$TSTNM,$BDADR,CS1,CS2,DS1,ER1
                                        DF10            ;0,0,0,0,0,0,0

                                ;*ITEM11
                                        EM11            ;WITH SPINDLE POWERED DOWN
                                                        ;RHCS2 SHOULD HAVE ONLY
```

```
1025
1026  001330  070071                    DH10          ;PC
1027                                                  ;TEST NO
1028                                                  ;FAILING REG. ADR
1029                                                  ;CONTENTS OF RHCS1
1030                                                  ;CONTENTS OF RHCS2
1031                                                  ;CONTENTS OF RHDS1
1032                                                  ;CONTENTS OF RHER1
1033  001332  071716                    DT10          ;$ERRPC,$TSTNM,$BDADR,CS1,CS2,DS1,ER1
1034  001334  072160                    DF10          ;0,0,0,0,0,0,0
1035
1036                              ;*ITEM12
1037  001336  052062                    EM12          ;AFTER A POWER UP WITH
1038                                                  ;NO PACK ACKNOWLEDGE COMMAND
1039                                                  ;RHDS1 SHOULD HAVE MOL=1, VV=0
1040  001340  070071                    DH10          ;PC
1041                                                  ;TEST NO
1042                                                  ;FAILING REGISTER ADDR.
1043                                                  ;CONTENTS OF RHCS1
1044                                                  ;CONTENTS OF RHCS2
1045                                                  ;CONTETNS OF RHDS1
1046                                                  ;CONTENTS OF RHER1
1047  001342  071716                    DT10          ;$ERRPC,$TSTNM,$BDADR,CS1,CS2,DS1,ER1
1048  001344  072160                    DF10          ;0,0,0,0,0,0,0
1049
1050                              ;*ITEM13
1051  001346  052170                    EM13          ;AFTER A POWER UP WITHOUT
1052                                                  ;ANY INIT RHCS1 SHOULD
1053                                                  ;HAVE GO=0, DVA=1, RDY=1
1054                                                  ;IE=0, DISREGARD
1055                                                  ;ALL OTHER BITS
1056  001350  070071                    DH10          ;PC
1057                                                  ;TEST NO
1058                                                  ;FAILING REGISTER ADDR.
1059                                                  ;CONTENTS OF RHCS1
1060                                                  ;CONTENTS OF RHCS2
1061                                                  ;CONTENTS OF RHDS1
1062                                                  ;CONTENTS OF RHER1
1063  001352  071716                    DT10          ;$ERRPC,$TSTNM,$BDADR,CS1,CS2,DS1,ER1
1064  001354  072160                    DF10          ;0,0,0,0,0,0,0
1065
1066                              ;*ITEM14
1067  001356  052312                    EM14          ;AFTER POWER UP RHCC
1068                                                  ;SHOULD BE=0
1069  001360  067613                    DH5           ;PC
1070                                                  ;REG. ADDR.
1071                                                  ;GOOD DATA
1072                                                  ;RECEIVED DATA
1073  001362  071660                    DT5           ;$ERRPC,REGADR,$GDDAT,$BDDAT
1074  001364  072144                    DF5           ;0,0,0,0
1075
1076                              ;*ITEM15
1077  001366  052365                    EM15          ;PACK ACKNOWLEDGE CAUSED
1078                                                  ;AN ERROR
1079                                                  ;GOOD DATA IS BEFORE COMMAND
1080                                                  ;RECEIVED DATA IS AFTER COMMAND
```

```
1081  001370  067613                    DH5                 ;PC
1082                                                        ;REG. ADDR.
1083                                                        ;GOOD DATA
1084                                                        ;RECEIVED DATA
1085  001372  071660                    DT5                 ;$ERRPC,REGADR,$GDDAT,$BDDAT
1086  001374  072144                    DF5                 ;0,0,0,0
1087
1088                            ;*ITEM16
1089  001376  052526                    EM16                ;GIVING A NO-OP COMMAND CAUSED
1090                                                        ;AN ERROR
1091                                                        ;GOOD DATA GIVES REGISTER
1092                                                        ;CONTENTS BEFORE COMMAND
1093                                                        ;RECEIVED DATA GIVES REGISTER
1094                                                        ;CONTENTS AFTER COMMAND
1095  001400  067613                    DH5                 ;PC
1096                                                        ;REG. ADDR.
1097                                                        ;GOOD DATA
1098                                                        ;RECEIVED DATA
1099  001402  071660                    DT5                 ;$ERRPC,REGADR,$GDDAT,$BDDAT
1100  001404  072144                    DF5                 ;0,0,0,0
1101
1102                            ;*ITEM17
1103  001406  052654                    EM17                ;DRIVE CLEAR COMMAND
1104                                                        ;CAUSED AN ERROR
1105                                                        ;GOOD DATA GIVES WHAT SHOULD
1106                                                        ;BE THERE
1107                                                        ;RECEIVED DATA GIVES WHAT WAS
1108                                                        ;THERE AFTER COMMAND
1109  001410  067613                    DH5                 ;PC
1110                                                        ;REG. ADDR.
1111                                                        ;GOOD DATA
1112                                                        ;RECEIVED DATA
1113  001412  071660                    DT5                 ;$ERRPC,REGADR,$GDDAT,$BDDAT
1114  001414  072144                    DF5                 ;0,0,0,0
1115
1116                            ;*ITEM20
1117  001416  053011                    EM20                ;READ-IN COMMAND GAVE AN ERROR
1118                                                        ;GOOD DATA HAS WHAT SHOULD BE THERE
1119                                                        ;RECEIVED DATA HAS WHAT WAS
1120                                                        ;AFTER COMMAND
1121  001420  067613                    DH5                 ;PC
1122                                                        ;REG. ADDR.
1123                                                        ;GOOD DATA
1124                                                        ;RECEIVED DATA
1125  001422  071660                    DT5                 ;$ERRPC,REGADR,$GDDAT,$BDDAT
1126  001424  072144                    DF5                 ;0,0,0,0
1127
1128
1129                            ;*ITEM 21
1130  001426  053155                    EM21                ;RHCS1 CONTENTS DURING
1131                                                        ;COMMAND WAS IN ERROR
1132  001430  067613                    DH5
1133  001432  071660                    DT5
1134  001434  072144                    DF5
1135
1136                            ;*ITEM 22
```

```
1137  001436  053230                    EM22            ;RHDS1 CONTENTS DURING
1138                                                    ;COMM ANS WAS IN ERROR
1139  001440  067613                    DH5
1140  001442  071660                    DT5
1141  001444  072144                    DF5
1142
1143                              ;*ITEM 23
1144  001446  053303                    EM23            ;UNLOAD COMMAND GAVE AN ERROR
1145                                                    ;GOOD DATA GIVES WHAT SHOULD
1146                                                    ;BE THERE
1147                                                    ;RECEIVED DATA GIVES WHAT WAS
1148                                                    ;THERE AFTER COMMAND
1149  001450  067613                    DH5
1150  001452  071660                    DT5
1151  001454  072144                    DF5
1152
1153                              ;*ITEM 24
1154  001456  053446                    EM24            ;OFFSET COMMAND CAUSED AN ERROR
1155                                                    ;GOOD DATA IS WHAT SHOULD BE THERE
1156                                                    ;RECEIVED DATA GIVES WHAT WAS THERE
1157                                                    ;AFTER AN OFFSET COMMAND
1158  001460  067613                    DH5
1159  001462  071660                    DT5
1160  001464  072144                    DF5
1161
1162                              ;*ITEM 25
1163  001466  053611                    EM25            ;RETURN TO CENTER LINE COMMAND
1164                                                    ;CAUSED AN ERROR
1165                                                    ;GOOD DATA GIVES WHAT SHOULD BE
1166                                                    ;THERE
1167                                                    ;RECEIVED DATA GIVES WHAT WAS
1168                                                    ;THERE AFTER COMMAND
1169  001470  067613                    DH5
1170  001472  071660                    DT5
1171  001474  072144                    DF5
1172
1173                              ;*ITEM 26
1174  001476  053773                    EM26            ;500 OFFSETS CAUSED AN ERROR
1175  001500  070250                    DH26            ;PC
1176                                                    ;CONT. OF RHCS1
1177                                                    ;CONT. OF RHCS2
1178                                                    ;CONT. OF RHDS1
1179                                                    ;CONT. OF RHER1
1180                                                    ;CONT. OF RHER2
1181                                                    ;CONT. OF RHER3
1182  001502  071736                    DT26            ;$ERRPC,CS1,CS2,DS1,ER1,ER2,ER3
1183  001504  072167                    DF26            ;0,0,0,0,0,0,0
1184
1185                              ;*ITEM 27
1186  001506  054063                    EM27            ;WRITE HEADER AND DATA
1187                                                    ;CAUSED IMPROPER REGISTER CHANGE
1188                                                    ;GOOD DATA GIVES WHAT
1189                                                    ;SHOULD BE THERE
1190                                                    ;RECEIVED DATA GIVES WHAT
1191                                                    ;WAS THERE AFTER COMMAND
1192  001510  067613                    DH5
```

```
1193  001512  071660                    DT5
1194  001514  072144                    DF5
1195
1196                                ;*ITEM 30
1197  001516  054301                    EM30          ;WRITE HEADER AND DATA
1198                                                  ;CHANGED WRITE FROM BUFFER
1199  001520  070447                    DH30          ;PC
1200                                                  ;WORD NO
1201                                                  ;GOOD DATA
1202                                                  ;BAD DATA
1203  001522  071760                    DT30          ;$ERRPC,ERWORD,$GDDAT,$BDDAT
1204  001524  072177                    DF30          ;0,0,0,0
1205
1206                                ;*ITEM 31
1207  001526  054361                    EM31          ;READ HEADER AND DATA CAUSED
1208                                                  ;IMPROPER REGISTER CHANGE
1209                                                  ;GOOD DATA HAS WHAT SHOULD
1210                                                  ;BE THERE
1211                                                  ;RECEIVED DATA GIVES WHAT
1212                                                  ;WAS THERE AFTER COMMAND
1213  001530  067613                    DH5
1214  001532  071660                    DT5
1215  001534  072144                    DF5
1216
1217                                ;*ITEM 32
1218  001536  054576                    EM32          ;WRITE HEADER AND DATA FOLLOWED
1219                                                  ;BY A READ HEADER AND DATA
1220                                                  ;CAUSED A READ/WRITE ERROR
1221  001540  070447                    DH30
1222  001542  071760                    DT30
1223  001544  072177                    DF30
1224
1225                                ;*ITEM 33
1226  001546  054705                    EM33          ;READ DATA CAUSED IMPROPER REGISTER
1227                                                  ;CHANGE
1228                                                  ;GOOD DATA GIVES WHAT SHOULD BE THERE
1229                                                  ;RECEIVED DATA GIVES WHAT WAS THERE AFTER
1230                                                  ;COMMAND
1231  001550  067613                    DH5
1232  001552  071660                    DT5
1233  001554  072144                    DF5
1234
1235                                ;*ITEM 34
1236  001556  055107                    EM34          ;READ DATA INCORRECT
1237  001560  070447                    DH30
1238  001562  071760                    DT30
1239  001564  072177                    DF30
1240
1241                                ;*ITEM 35
1242  001566  055133                    EM35          ;WRITE DATA COMMAND CAUSED
1243                                                  ;IMPROPER REGISTER CHANGE
1244                                                  ;GOOD DATA GIVES WHAT SHOULD BE THERE
1245                                                  ;RECEIVED DATA GIVES REGISTER
1246                                                  ;CONTENTS AFTER WRITE DATA
1247  001570  067613                    DH5
1248  001572  071660                    DT5
```

```
1249  001574  072144                          DF5
1250
1251                                  ;*ITEM 36
1252  001576  055351                          EM36            ;WRITE DATA COMMAND CHANGED
1253                                                          ;WRITE FROM BUFFER
1254  001600  070447                          DH30
1255  001602  071760                          DT30
1256  001604  072177                          DF30
1257
1258                                  ;*ITEM 37
1259  001606  055426                          EM37            ;SEEK COMMAND CAUSED AN
1260                                                          ;ERROR
1261                                                          ;GOOD DATA GIVES WHAT SHOULD
1262                                                          ;BE THERE
1263                                                          ;RECEIVED DATA GIVES WHAT
1264                                                          ;WAS THERE AFTER SEEK COMMAND
1265  001610  067613                          DH5             ;
1266  001612  071660                          DT5             ;
1267  001614  072144                          DF5             ;
1268
1269                                  ;*ITEM 40
1270  001616  055643                          EM40            ;WRITE CHECK CAUSED AN
1271                                                          ;IMPROPER REGISTER CHANGE
1272                                                          ;GOOD DATA GIVES WHAT SHOULD
1273                                                          ;BE THERE
1274                                                          ;RECEIVED DATA GIVES WHAT WAS
1275                                                          ;THERE AFTER COMMAND
1276  001620  067613                          DH5
1277  001622  071660                          DT5
1278  001624  072144                          DF5
1279
1280                                  ;*ITEM 41
1281  001626  056052                          EM41            ;LOCKING OUT WRITES BY WRITE
1282                                                          ;LOCK BUTTON CAUSED IMPROPER
1283                                                          ;REGISTER CHANGE
1284                                                          ;GOOD DATA GIVES WHAT SHOULD
1285                                                          ;BE THERE
1286                                                          ;RECEIVED DATA GIVES WHAT
1287                                                          ;WAS THERE AFTER WRITES
1288                                                          ;WERE LOCKED OUT BY
1289                                                          ;BUTTON
1290  001630  067613                          DH5
1291  001632  071660                          DT5
1292  001634  072144                          DF5
1293
1294                                  ;*ITEM 42
1295  001636  056333                          EM42            ;ATTEMPTING TO WRITE WITH WRITE
1296                                                          ;LOCKED OUT CAUSED IMPROPER
1297                                                          ;REGISTER CHANGE
1298                                                          ;GOOD DATA GIVES WHAT SHOULD
1299                                                          ;BE THERE
1300                                                          ;RECEIVED DATA GIVES WHAT WAS
1301                                                          ;THERE AFTER ATTEMPT
1302  001640  067613                          DH5
1303  001642  071660                          DT5
1304  001644  072144                          DF5
```

```
1305
1306                                    ;*ITEM 43
1307  001646  056611                          EM43                    ;WRITING WITH WRITE LOCKED
1308                                                                    ;OUT CHANGED DISK DATA
1309                                                                    ;GOOD DATA GIVES WHAT WAS
1310                                                                    ;ON DISK BEFORE WRITE WITH
1311                                                                    ;WRITE LOCK WAS ATTEMPTED
1312                                                                    ;RECEIVED DATA GIVES WHAT WAS
1313                                                                    ;READ BACK AFTER WRITE WITH
1314                                                                    ;WRITE LOCK WAS ATTEMPTED
1315  001650  070447                            DH30
1316  001652  071760                            DT30
1317  001654  072177                            DF30
1318
1319                                    ;*ITEM 44
1320  001656  057153                          EM44                    ;ENABLING WRITES BY WRITE LOCK
1321                                                                    ;BUTTON CAUSED AN ERROR
1322                                                                    ;GOOD DATA GIVES WHAT SHOULD
1323                                                                    ;BE THERE
1324                                                                    ;RECEIVED DATA GIVES WHAT WAS
1325                                                                    ;THERE AFTER WRITE LOCK
1326                                                                    ;BUTTON ENABLED WRITES
1327  001660  067613                            DH5                     ;
1328  001662  071660                            DT5                     ;
1329  001664  072144                            DF5                     ;
1330
1331                                    ;*ITEM 45
1332  001666  057445                          EM45                    ;TRANSFERRING ON LAST BLOCK IE. CYLINDER
1333                                                                    ;410, SECTOR 21, TRACK 18
1334                                                                    ;CAUSED IMPROPER REGISTER
1335                                                                    ;CHANGE
1336                                                                    ;GOOD DATA GIVES WHAT SHOULD
1337                                                                    ;BE THERE
1338                                                                    ;RECEIVED DATA GIVES WHAT WAS
1339                                                                    ;THERE AFTER TRANSFER
1340  001670  067613                            DH5
1341  001672  071660                            DT5
1342  001674  072144                            DF5
1343
1344                                    ;*ITEM 46
1345  001676  057741                          EM46                    ;DATA READ FROM LAST
1346                                                                    ;BLOCK IE. CYLINDER 410
1347                                                                    ;SECTOR 21, TRACK 18 IS IN
1348                                                                    ;ERROR
1349  001700  070447                            DH30
1350  001702  071760                            DT30
1351  001704  072177                            DF30
1352
1353                                    ;*ITEM 47
1354  001706  060054                          EM47                    ;TRANSFERRING FROM NONEXISTANT
1355                                                                    ;SECTOR CAUSED IMPROPER
1356                                                                    ;REGISTER CHANGE
1357                                                                    ;GOOD DATA GIVES WHAT SHOULD
1358                                                                    ;BE THERE
1359                                                                    ;RECEIVED DATA GIVES WHAT WAS
1360                                                                    ;THERE AFTER ATTEMPTED
```

```
1361
1362  001710  067613                        DH5                     ;TRANSFER
1363  001712  071660                        DT5
1364  001714  072144                        DF5
1365
1366                                ;*ITEM 50
1367  001716  060336                        EM50                    ;TRANSFERRING FROM NONEXISTANT
1368                                                                  ;SECTOR CAUSED DATA ERROR
1369                                                                  ;GOOD DATA GIVES WHAT
1370                                                                  ;SHOULD BE IN BUFFER
1371                                                                  ;RECEIVED DATA GIVES WHAT WAS
1372                                                                  ;IN BUFFER AFTER TRANSFER
1373  001720  070447                        DH30
1374  001722  071760                        DT30
1375  001724  072177                        DF30
1376
1377                                ;*ITEM 51
1378  001726  060555                        EM51                    ;GIVING ILLEGAL FUNCTION CAUSED
1379                                                                  ;IMPROPER REGISTER CHANGE
1380                                                                  ;GOOD DATA GIVES WHAT SHOULD BE
1381                                                                  ;THERE
1382                                                                  ;RECEIVED DATA GIVES REGISTER
1383                                                                  ;CONTENTS AFTER ILLEGAL FUNCTION
1384  001730  070563                        DH51                    ;PC
1385                                                                  ;REG. ADDR.
1386                                                                  ;GOOD DATA
1387                                                                  ;RECEIVED DATA
1388                                                                  ;ILLEGAL FUNCTION
1389  001732  071774                        DT51                    ;$ERRPC,REGADR,$GDDAT,$BDDAT,ILLEGL
1390  001734  072204                        DF51                    ;0,0,0,0,0
1391
1392
1393                                ;*ITEM 52
1394  001736  061022                        EM52                    ;WRITE DATA ON NONEXISTANT
1395                                                                  ;SECTOR CAUSED IMPROPER
1396                                                                  ;REGISTER CHANGE
1397                                                                  ;GOOD DATA GIVES WHAT SHOULD
1398                                                                  ;BE THERE
1399                                                                  ;RECEIVED DATA GIVES WHAT
1400                                                                  ;WAS THERE AFTER ATTEMPTED
1401                                                                  ;WRITE DATA
1402  001740  067613                        DH5
1403  001742  071660                        DT5
1404  001744  072144                        DF5
1405
1406                                ;*ITEM 53
1407  001746  061277                        EM53                    ;READ HEADER AND DATA AFTER
1408                                                                  ;A SEARCH CAUSED AN ERROR
1409  001750  070447                        DH30
1410  001752  071760                        DT30
1411  001754  072177                        DF30
1412
1413                                ;*ITEM 54
1414  001756  061365                        EM54                    ;ATTEMPTED OPERATION WITH
1415                                                                  ;INVALID ADDRESS CAUSED
1416                                                                  ;IMPROPER REGISTER CHANGE
```

```
                                                        ;GOOD DATA GIVES WHAT SHOULD
                                                        ;BE THERE
                                                        ;RECEIVED DATA GIVES WHAT WAS
                                                        ;THERE AFTER OPERATION
001760  067613                        DH5
001762  071660                        DT5
001764  072144                        DF5

                                ;*ITEM 55
001766  061632                        EM55            ;WRITING/READING WITH EXPECTED
                                                        ;ADDRESS OVERFLOW ERROR CAUSED
                                                        ;IMPROPER REGISTER CHANGE
                                                        ;GOOD DATA GIVES WHAT SHOULD
                                                        ;BE THERE
                                                        ;RECEIVED DATA GIVES WHAT
                                                        ;WAS THERE AFTER OPERATION
001770  067613                        DH5
001772  071660                        DT5
001774  072144                        DF5

                                ;*ITEM 56
001776  062120                        EM56            ;DATA READ WITH AN EXPECTED
                                                        ;ADDRESS OVERFLOW ERROR IS
                                                        ;INCORRECT
                                                        ;WORD NO 1 TO 260 SHOULD
                                                        ;BE READ
                                                        ;WORD NOS 261 TO 266 SHOULD
                                                        ;NOT CHANGE DUE TO READ
002000  070447                          DH30
002002  071760                          DT30
002004  072177                          DF30

                                ;*ITEM 57
002006  062330                        EM57            ;ATTEMPTING DATA COMMAND
                                                        ;WITH WRONG FORMAT BIT CAUSED
                                                        ;IMPROPER REGISTER CHANGE
                                                        ;GOOD DATA GIVES WHAT SHOULD BE
                                                        ;THERE
                                                        ;RECEIVED DATA GIVES WHAT WAS
                                                        ;THERE AFTER ATTEMPTED DATA
                                                        ;TRANSFER
002010  067613                        DH5
002012  071660                        DT5
002014  072144                        DF5

                                ;*ITEM 60
002016  062622                        EM60            ;ATTEMPTING TO MODIFY REGISTER
                                                        ;DURING AN OPERATION CAUSED
                                                        ;IMPROPER REGISTER CHANGE
                                                        ;GOOD DATA GIVES WHAT SHOULD
                                                        ;BE THERE
                                                        ;RECEIVED DATA GIVES WHAT WAS
                                                        ;THERE AFTER OPERATION
                                                        ;WAS COMPLETE
002020  070722                          DH60            ;PC
                                                        ;REG. ADDR.
```

```
1473                                                    ;GOOD DATA
1474                                                    ;RECEIVED DATA
1475                                                    ;MODFING REGISTER
1476  002022  072012                    DT60           ;$ERRPC,REGADR,$GDDAT,$BDDAT,$BDADR
1477  002024  072212                    DF60           ;0,0,0,0,0
1478
1479                              ;*ITEM 61
1480  002026  063233                    EM61           ;DEVICE NOT AVAILBLE BEFORE COMMAND WAS TO BE GIVEN
1481  002030  071057                    DH61           ;PC
1482                                                    ;TEST NO.
1483                                                    ;PC OF JSR
1484                                                    ;RHCS1 CONTENTS
1485  002032  072030                    DT61           ;$ERRPC,$TSTNM,PCJSR,$BDADR
1486  002034  072220                    DF61           ;0,0,0,0
1487
1488                              ;*ITEM 62
1489  002036  063314                    EM62           ;RHDS1 HAS STATUS BITS STUCK AT ONE
1490  002040  071152                    DH62           ;PC
1491                                                    ;TEST NO.
1492                                                    ;PC OF JSR
1493                                                    ;RHDS1 CONTENTS
1494  002042  072042                    DT62           ;$ERRPC,$TSTNM,PCJSR,$BDADR
1495  002044  072224                    DF62           ;0,0,0,0
1496
1497
1498                              ;*ITEM 63
1499  002046  063375                    EM63           ;RHDS1 CONTENTS DURING
1500                                                    ;COMMAND WAS IN ERROR
1501  002050  067613                    DH5
1502  002052  071660                    DT5
1503  002054  072144                    DF5
1504
1505
1506                              ;*ITEM 64
1507  002056  063451                    EM64           ;RECALIBRATE COMMAND CAUSED
1508                                                    ;IMPROPER REGISTER CHANGE.
1509                                                    ;GOOD DATA GIVES WHAT SHOULD BE
1510                                                    ;THERE.
1511                                                    ;RECEIVED DATA GIVES WHAT WAS THERE
1512                                                    ;AFTER COMMAND
1513  002060  067613                    DH5
1514  002062  071660                    DT5
1515  002064  072144                    DF5
1516
1517
1518                              ;*ITEM65
1519
1520  002066  063670                    EM65           ;INTERRUPT FAILING
1521  002070  071245                    DH65           ;PC
1522                                                    ;TEST NO
1523                                                    ;CONTENTS OF RHCS1
1524                                                    ;CONTENTS OF RHAS
1525                                                    ;CONTENTS OF RHDS1
1526  002072  072054                    DT65           ;$ERRPC,TSTNM,CS1,AS,DS1
1527  002074  072230                    DF65           ;0,0,0,0,0
1528
```

```
1529
1530                                    ;*ITEM66
1531  002076  063712                          EM66            ;HEADER AND DATA COMMAND
1532                                                          ;FOR HEAD SELECTION TEST
1533                                                          ;CAUSED AN ERROR
1534                                                          ;RHDST GIVES WHAT TRACK
1535                                                          ;WAS BEING WRITTEN ON CYLINDER 0
1536                                                          ;SECTOR 0
1537  002100  071370                          DH66            ;PC
1538                                                          ;RHDST
1539                                                          ;RHER1
1540                                                          ;RHER2
1541                                                          ;RHER3
1542                                                          ;RHCS1
1543                                                          ;RHCS2
1544  002102  072070                          DT66            ;$ERRPC,DST,ER1,ER2,ER3,CS1,CS2
1545  002104  072235                          DF66            ;0,0,0,0,0,0,0
1546                                    ;*ITEM67
1547  002106  064104                          EM67            ;READ HEADER AND DATA ERROR
1548                                                          ;IN HEAD SELECTION TEST
1549                                                          ;FIRST FOUR WORDS GIVE HEADER
1550                                                          ;NEXT WORDS ARE DATA
1551                                                          ;GOOD DATA WORDS GIVE
1552                                                          ;THE TRACK NUMBER IN
1553                                                          ;BITS 4,5,6,7,8
1554  002110  070447                          DH30
1555  002112  071760                          DT30
1556  002114  072177                          DF30
1557                                    ;*ITEM70
1558  002116  064374                          EM70            ;READ HEADER AND DATA ERROR
1559                                                          ;IN DIFFERENCE LINE TEST
1560                                                          ;WORD NOS. 1-4 GIVE
1561                                                          ;HEADER
1562                                                          ;WORD NOS. 5-260 GIVE DATA
1563                                                          ;WHICH IS THE CYLINDER
1564                                                          ;ADDRESS
1565  002120  070447                          DH30
1566  002122  071760                          DT30
1567  002124  072177                          DF30
1568
1569                                    ;*ITEM 71
1570  002126  064602                          EM71            ;FORCING OPI CAUSED IMPROPER REGISTER
1571                                                          ; CHANGE
1572                                                          ;GOOD DATA GIVES WHAT SHOULD
1573                                                          ;BE THERE
1574                                                          ;RECEIVED DATA GIVES WHAT WAS
1575                                                          ;THERE AFTER 3 INDEX PULSES
1576  002130  067613                          DH5             ;PC
1577                                                          ;REG. ADDR.
1578                                                          ;GOOD DATA
1579                                                          ;RECEIVED DATA
1580  002132  071660                          DT5             ;$ERRPC,REGADR,$GDDAT,$BDDAT
1581  002134  072144                          DF5             ;0,0,0,0
1582                                    ;*ITEM 74
1583  002136  065345                          EM74            ;WHILE USING UNIBUS B
1584                                                          ;READ DATA CAUSED IMPROPER REGISTER
```

```
1585                                                    ;CHANGE
1586                                                    ;GOOD DATA GIVES WHAT SHOULD BE THERE
1587                                                    ;RECEIVED DATA GIVES WHAT WAS THERE AFTER
1588                                                    ;COMMAND
1589  002140  067613                    DH5
1590  002142  071660                    DT5
1591  002144  072144                    DF5
1592
1593                              ;*ITEM 73
1594  002146  065273                    EM73           ;WHILE USING UNIBUS B
1595                                                    ;READ DATA INCORRECT
1596  002150  070447                    DH30
1597  002152  071760                    DT30
1598  002154  072177                    DF30
1599
1600                              ;*ITEM 74
1601  002156  065345                    EM74           ;WHILE USING UNIBUS B
1602                                                    ;WRITE DATA COMMAND CAUSED
1603                                                    ;IMPROPER REGISTER CHANGE
1604                                                    ;GOOD DATA GIVES WHAT SHOULD BE THERE
1605                                                    ;RECEIVED DATA GIVES REGISTER
1606                                                    ;CONTENTS AFTER WRITE DATA
1607  002160  067613                    DH5
1608  002162  071660                    DT5
1609  002164  072144                    DF5
1610
1611                              ;*ITEM 75
1612  002166  065611                    EM75           ;WHILE USING UNIBUS B
1613                                                    ;WRITE DATA COMMAND CHANGED
1614                                                    ;WRITE FROM BUFFER
1615  002170  070447                    DH30
1616  002172  071760                    DT30
1617  002174  072177                    DF30
1618
1619                              ;*ITEM 76
1620  002176  065714                    EM76           ;WHILE USING UNIBUS B
1621                                                    ;WRITE CHECK CAUSED AN
1622                                                    ;IMPROPER REGISTER CHANGE
1623                                                    ;GOOD DATA GIVES WHAT SHOULD
1624                                                    ;BE THERE
1625                                                    ;RECEIVED DATA GIVES WHAT WAS
1626                                                    ;THERE AFTER COMMAND
1627  002200  067613                    DH5
1628  002202  071660                    DT5
1629  002204  072144                    DF5
1630
1631                              ;*ITEM 77
1632  002206  066151                    EM77           ;CURRENT CYLINDER DOES NOT REFLECT DESIRED 'RHCC'
1633  002210  071466                    DH77           ;PC
1634                                                    ;PC OF JSR
1635                                                    ;REGISTER ADDRESS
1636                                                    ;GOOD DATA
1637                                                    ;BAD DATA
1638  002212  072110                    DT77           ;$ERRPC,PCJSR,REGADR,$GDDAT,$BDDAT
1639  002214  072245                    DF77           ;0,0,0,0,0
1640
```

```
1641                                    ;*ITEM 100
1642  002216  066374                            EM100           ;ERROR AFTER ADDRESS PLUG CHANGE
1643  002220  067613                            DH5             ;PC
1644                                                            ;REGISTER ADDRESS
1645                                                            ;GOOD DATA
1646                                                            ;RECEIVED DATA
1647  002222  071660                            DT5             ;$ERRPC,REGADR,$GDDAT,$BDDAT
1648  002224  072144                            DF5             ;0,0,0,0
1649
1650                                    ;*ITEM 101
1651  002226  066456                            EM101           ;UNIT DID NOT GO OFFLINE WHEN ADDR
1652                                                            ;PLUG WAS REMOVED
1653  002230  070250                            DH26            ;PC
1654                                                            ;CONT OF RHCS1
1655                                                            ;CONT OF RHCS2
1656                                                            ;CONT OF RHDS1
1657                                                            ;CONT OF RHER2
1658                                                            ;CONT OF RHER3
1659  002232  071736                            DT26            ;$ERRPC,CS1,CS2,DS1,ER2,ER3
1660  002234  072167                            DF26            ;0,0,0,0,0,0,0
1661
1662                                    ;*ITEM 102
1663  002236  066540                            EM102           ;UNIT DID NOT COME BACK ONLINE WHEN
1664                                                            ;ADDR PLUG WAS REPLACED
1665  002240  070250                            DH26            ;PC
1666                                                            ;CONT OF RHCS1
1667                                                            ;CONT OF RHCS2
1668                                                            ;CONT OF RHDS1
1669                                                            ;CONT OF RHER2
1670                                                            ;CONT OF RHER3
1671  002242  071736                            DT26            ;$ERRPC,CS1,CS2,DS1,ER2,ER3
1672  002244  072167                            DF26            ;0,0,0,0,0,0,0
1673
1674                                    ;*ITEM  103
1675  002246  066617                            EM103           ;REGISTER CONTENTS INCORRECT BEFORE A
1676                                                            ;DIAGNOSTIC SEEK
1677  002250  070250                            DH26            ;PC
1678                                                            ;CONT OF RHCS1
1679                                                            ;CONT OF RHCS2
1680                                                            ;CONT OF RHDS1
1681                                                            ;CONT OF RHER2
1682                                                            ;CONT OF RHER3
1683  002252  071736                            DT26            ;$ERRPC,CS1,CS2,DS1,ER2,ER3
1684  002254  072167                            DF26            ;0,0,0,0,0,0,0
1685
1686                                    ;*ITEM  104
1687  002256  066703                            EM104           ;REGISTER CONTENTS INCORRECT AFTER A
1688                                                            ;DIAGNOSTIC SEEK
1689  002260  070250                            DH26            ;PC
1690                                                            ;CONT OF RHCS1
1691                                                            ;CONT OF RHCS2
1692                                                            ;CONT OF RHDS1
1693                                                            ;CONT OF RHER2
1694                                                            ;CONT OF RHER3
1695  002262  071736                            DT26            ;$ERRPC,CS1,CS2,DS1,ER2,ER3
1696  002264  072167                            DF26            ;0,0,0,0,0,0,0
```

```
1697
```

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


002266  000254                  RPVEC:  254                     ;RP04 VECTOR ADDRESS


                                ;*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                          UNIB=   20                      ;UNIBUS B DC LO (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)
020000                  MCPE=   20000           ;MASSBUSS PARITY ERROR (BIT #13)
040000                  TRE=    40000           ;TRANSFER ERROR (BIT #14)
100000                  SC=     100000          ;SPECIAL CONDITION (BIT #15)

                        ;*STATUS REGISTER (RHDS1) (#01)

000001                  DFF5=   1               ;DRIVE FORWARD 5''/SEC.  (BIT #0)
000002                  DFF20=  2               ;DRIVE FORWARD 20''/SEC.  (BIT #1)
000004                  DIGB=   4               ;DRIVE TO INNER GAVRD BAND (BIT #2)
000010                  GRV=    10              ;GO REVERSE (BIT #3)
000020                  DL64=   20              ;DIFFERENCE LESS THAN 64 (BIT #4)
000040                  DE1=    40              ;DIFFERENCE EQUALS 1 (BIT #5)
000100                  VV=     100             ;VOLUME VALID (BIT #6)
000200                  DRY=    200             ;DRIVE READY (BIT #7)
000400                  DPR=    400             ;DRIVE PRESENT (BIT #8)
001000                  PROG=   1000            ;PROGRAMABLE (BIT #9)
002000                  LBT=    2000            ;LAST SECTOR TRANSFERRED (BIT #10)
004000                  WRL=    4000            ;WRITE LOCK (BIT #11)
010000                  MOL=    10000           ;MEDIUM ON-LINE (BIT #12)
020000                  PIP=    20000           ;POSITIONING OPERATION IN PROGRESS (BIT #13)
040000                  ERR=    40000           ;COMPOSIT ERROR. (BIT #14)
100000                  ATA=    100000          ;ATTENTION ACTIVE (BIT #15)

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

```
                        ;*MAINTAINABILITY REGISTER (RHMR)(#03)

000001                  DMD=    1               ;DIAGINOSTIC MODE (BIT #0)
000002                  MCLK=   2               ;MAINTAINABILITY CLOCK (BIT #1)
000004                  MINX=   4               ;MAINTAINABILITY INDEX (BIT #2)
000010                  MSTCK=  10              ;MAINTAINABILITY SECTOR CLOCK (BIT #3)
000020                  MRD=    20              ;MAINTAINABILITY READ (BIT #4)
000040                  MWR=    40              ;MAINTAINABILITY WRITE (BIT #5)
001000                  DTSY=   1000            ;MAINTAINABILITY SYNC DETECTED (BIT #9)

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

000001                  AT0=    1               ;DEVICE 0 (BIT #0)
000002                  AT1=    2               ;DEVICE 1 (BIT #1)
000004                  AT2=    4               ;DEVICE 2 (BIT #2)
000010                  AT3=    10              ;DEVICE 3 (BIT #3)
000020                  AT4=    20              ;DEVICE 4 (BIT #4)
000040                  AT5=    40              ;DEVICE 5 (BIT #5)
000100                  AT6=    100             ;DEVICE 6 (BIT #6)
000200                  AT7=    200             ;DEVICE 7 (BIT #7)

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




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




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

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

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

```
1853    000001          WCU=    1               ;WRITE CURRENT UNSAFE (BIT #0)
1854    000002          CSF=    2               ;CURRENT SINK FAILURE (BIT #1)
1855    000004          WSU=    4               ;WRITE SELECT UNSAFE (BIT #2)
1856    000010          CSU=    10              ;CURRENT SWITCH UNSAFE (BIT #3)
1857    000020          MSE=    20              ;MOTOR SEQUENCE ERROR (BIT #4)
1858    000040          TDF=    40              ;TRANSITIONS DETECTOR FAILURE (BIT #5)
1859    000100          TUF=    100             ;TRANSITIONS UNSAFE (BIT #6)
1860    000200          FEN=    200             ;FAILSAFE ENABLED (BIT #7)
1861    000400          WRU=    400             ;WRITE READY UNSAFE (BIT #8)
1862    001000          MHS=    1000            ;MULTIPLE HEAD SELECT (BIT #9)
1863    002000          NHS=    2000            ;NO HEAD SELECTION (BIT #10)
1864    004000          IXE=    4000            ;INDEX ERROR (BIT #11)
1865    010000          VU30=   10000           ;30VOLT UNSAFE (BIT #12)
1866    020000          PLU=    20000           ;PLO UNSAFE (BIT #13)
1867    100000          ACU=    100000          ;ACUNSAFE (BIT #15)
1868
1869                    ;*OFFSET REGISTER (RHOF) (#11)
1870
1871    000001          OF25=   1               ;OFFSET 25 MICRO INCHES (BIT #0)
1872    000002          OF50=   2               ;OFFSET 50 MICRO INCHES (BIT #1)
1873    000004          OF100=  4               ;OFFSET 100 MICRO INCHES (BIT #2)
1874    000010          OF200=  10              ;OFFSET 200 MICRO INCHES (BIT #3)
1875    000020          OF400=  20              ;OFFSET 400 MICRO INCHES (BIT #4)
1876    000040          OF800=  40              ;OFFSET 800 MICRO INCHES (BIT #5)
1877
1878    000200          OFREV=  200             ;OFFSET NEGATIVE (REVERSE) (BIT #5)
1879    002000          HCI=    2000            ;HEADER COMPARE INHIBIT (BIT #10)
1880    004000          ECI=    4000            ;ERROR CORRECTION CODE INHIBIT (BIT #11)
1881    010000          FMT22=  10000           ;FORMAT BIT (BIT #12
1882
1883                    ;*DESIRED CYLINDER ADDRESS (RHCA) (#12)
1884                    ;*EACH BIT IS CALLED BY BIT NUMBER.
1885
1886
1887
1888
1889                    ;*CURRENT CYLINDER ADDRESS (RHCC) (#13)
1890                    ;*EACH BIT IS CALLED BY BIT NUMBER
1891
1892
1893
1894
1895                    ;*SERIAL NUMBER REGISTER (RHSN) (#14)
1896                    ;*EACH IS CALLED BY BIT NUMBER
1897
1898
1899
1900
1901                    ;*ERROR REGISTER #03 (RHER3) (#15)
1902
1903    000001          PSU=    1               ;PACK SPEED UNSAFE (BIT #0)
1904    000002          VUF=    2               ;VELOCITY UNSAFE (BIT #1)
1905    000010          UWR=    10              ;ANY UNSAFE EXCEPT READ/WRITE (BIT #3)
1906    000020          PRE=    20              ;DISK PACK ROTATION ERROR (BIT #4)
1907    000040          ACL=    40              ;AC LOW (BIT #5)
1908    000100          DCL=    100             ;DC LOW (BIT #6)
```

```
1909            020000                  ACE=    20000                   ;ADDRESS CHANGE ERROR (BIT #13)
1910            040000                  SKI=    40000                   ;SEEK INCOMPLETE (BIT #14)
1911            100000                  OCYL=   100000                  ;OFF CYLINDER (BIT #15)
1912
1913
1914
1915                                    ;*ECC POSITION REGISTER (RHEC1) (#16)
1916                                    ;*EACH BIT IS CALLED BY BIT NUMBER
1917
1918
1919
1920
1921                                    ;*ECC PATTERN REGISTER (RHEC2) (#17)
1922                                    ;*EACH BIT IS CALLED BY BIT NUMBER
1923
1924
1925
1926
1927
1928
1929
1930
1931
```

```
1932                                    .SBTTL  REGISTER ADDRESSES
1933
1934
1935
1936
1937                                    ;*RP04/5/6 DISK I/O REGISTER LOCATED IN THE RH11 CONTROLLER
1938
1939  002270  176722                    RHDB:   176722          ;DATA BUFFER
1940  002272  176702                    RHWC:   176702          ;WORD COUNT
1941  002274  176704                    RHBA:   176704          ;BUS ADDRESS
1942  002276  176710                    RHCS2:  176710          ;CONTROL AND STATUS 2
1943
1944                                    ;*RP04/5/6 DISK I/O REGISTERS LOCATED IN THE DEVICE CONTROL LOGIC (DCL)
1945
1946  002300  176700                    RHCS1:  176700          ;CONTROL AND STATUS 1
1947  002302  176714                    RHER1:  176714          ;ERROR #1
1948  002304  176706                    RHDST:  176706          ;DESIRED SECTOR/TRACK ADDRESS
1949  002306  176740                    RHER2:  176740          ;ERROR #2
1950  002310  176732                    RHOF:   176732          ;OFFSET
1951  002312  176734                    RHCA:   176734          ;DESIRED CYLINDER ADDRESS
1952  002314  176742                    RHER3:  176742          ;ERROR #3
1953  002316  176716                    RHAS:   176716          ;ATTENTION SUMMARY
1954  002320  176724                    RHMR:   176724          ;MAINTAINABILITY
1955  002322  176712                    RHDS1:  176712          ;DRIVE STATUS
1956  002324  176726                    RHDT:   176726          ;DRIVE TYPE
1957  002326  176730                    RHSN:   176730          ;SERIAL NUMBER
1958  002330  176744                    RHEC1:  176744          ;ECC POSITION
1959  002332  176746                    RHEC2:  176746          ;ECC PATTERN
1960  002334  176736                    RHCC:   176736          ;CURRENT CYLINDER ADDRESS
1961  002336  176720                    RHLA:   176720          ;LOOK-AHEAD
1962
1963                                    ;*ADDITIONAL I/O REGISTERS LOCATED IN THE RH70 CONTROLLER LOGIC
1964
1965  002340  176750                    RHBAE:  176750          ;BUS ADDRESS EXTENSION REGISTER
1966  002342  176752                    RHCS3:  176752          ;CONTROL AND STATUS REGISTER #3
1967
1968
1969                                    ;*P-CLOCK (KW11-P) I/O REGISTERS
1970
1971  002344  172540                    PCLCSR: 172540          ;CONTROL AND STATUS REGISTER
1972  002346  172542                    PCLBUF: 172542          ;COUNT SET BUFFER
1973  002350  172544                    PCLCTR: 172544          ;COUNTER
1974
```

```
1975
1976
1977                            ;*THE FOLLOWING LOCATIONS ARE RESERVED FOR REGISTER SNAPSHOTS
1978                            ;*ANY TIME THERE IS AN ERROR ALL THESE WILL BE FILLED
1979
1980                            ;*ONLY SOME MAY BE PRINTED BUT ALL WILL BE FILLED TRUE
1981                            ;*FOR THE TIME JUST AFTER THE "ERROR" ERROR COMMAND
1982
1983
1984                            ;;*****************************************************************
1985                            ;*THIS ASSUMES THAT A REGISTER SNAPSHOT HAS BEEN TAKEN WHICH IS NOT
1986                            ;*ALWAYS THE CASE - IF QUESTIONABLE CONTENTS APPEAR IN THE REGISTER
1987                            ;*PRINTOUTS, CHECK THE INLINE TEST CODE TO SEE IF THE REGISTER SNAPSHOT
1988                            ;*REFLECTS THE CURRENT STATE OF THE MACHINE
1989                            ;;*****************************************************************
1990
1991
1992  002352  000000            DB:     0                       ;DATA BUFFER
1993  002354  000000            WC:     0                       ;WORD COUNT
1994  002356  000000            BA:     0                       ;BUS ADDRESS
1995  002360  000000            CS2:    0                       ;CONTROL AND STATUS 2
1996
1997
1998  002362  000000            CS1:    0                       ;CONTROL AND STATUS 1
1999  002364  000000            ER1:    0                       ;ERROR #1
2000  002366  000000            DST:    0                       ;DESIRED SECTOR/TRACK ADDRESS
2001  002370  000000            ER2:    0                       ;ERROR #2
2002  002372  000000            OF:     0                       ;OFFSET
2003  002374  000000            CA:     0                       ;DESIRED CYLINDER ADDRESS
2004  002376  000000            ER3:    0                       ;ERROR #3
2005  002400  000000            AS:     0                       ;ATTENTION SUMMARY
2006  002402  000000            MR:     0                       ;MAINTAINABILITY
2007  002404  000000            DS1:    0                       ;DRIVE STATUS
2008  002406  000000            DT:     0                       ;DRIVE TYPE
2009  002410  000000            SN:     0                       ;SERIAL NUMBER
2010  002412  000000            EC1:    0                       ;ECC POSITION
2011  002414  000000            EC2:    0                       ;ECC PATTERN
2012  002416  000000            CC:     0                       ;CURRENT CYLINDER ADDRESS
2013  002420  000000            LA:     0                       ;LOOK-AHEAD
2014
```

```
2015
2016
2017                                    ;*FUNCTION EQUATES
2018
2019                                        ;*TABLE OF FUNCTIONS FOR RHCS1 THEN "GO" BIT HAS TO BE SET
2020  002422                            FUTABL:
2021  002422  000000                    NOPERA: 0                       ;NO OPERATION
2022  002424  000002                    UNLOAD: 2                       ;UNLOAD (STAND BY)
2023  002426  000006                    RECALI: 6                       ;RECALIBRATE
2024  002430  000010                    DCLEAR: 10                      ;DRIVE CLEAR
2025  002432  000012                    RELEAS: 12                      ;RELEASE (DUAL-PORT OPERATION)
2026  002434  000030                    SERCH:  30                      ;SEARCH COMMAND
2027  002436  000050                    WRCHEK: 50                      ;WRITE CHECK DATA
2028  002440  000052                    WRCHDT: 52                      ;WRITE CHECK HEADER AND DATA
2029  002442  000060                    WRIDAT: 60                      ;WRITE DATA
2030  002444  000062                    WRIFOR: 62                      ;WRITE HEADER AND DATA (FORMAT)
2031  002446  000070                    READAT: 70                      ;READ DATA
2032  002450  000072                    REFOR:  72                      ;READ HEADER AND DATA
2033  002452  000004                    SEECOM: 4                       ;SEEK COMMAND
2034  002454  000014                    OFSETC: 14                      ;OFFSET COMMAND
2035  002456  000016                    RETCL:  16                      ;RETURNTO CENTERLINE
2036  002460  000022                    PKACK:  22                      ;PACK ACKNOWLEDGE
2037  002462  000020                    READIN: 20                      ;READ IN
2038  002464  000000                    ILLEGL: .WORD   0               ;COMPUTED ILLEGAL FUNCTION
2039
2040
2041
2042                                    ;*DATA BUFFERS FOR READ/WRITE
2043
2044
2045  002470  000422                    WRFROM: .BLKW   274.            ;WRITE FROM THIS BUFFER
2046  003534  000422                    REINTO: .BLKW   274.            ;READ INTO THIS BUFFER
2047
```

```
2048
2049
2050                                    ;*RESERVED CORE LOCATIONS
2051
2052  004600  000000                    REGADR: 0                       ;SAVE REGISTER ADDRESS HERE
2053  004602  000000                    ERWORD: 0                       ;SAVE ERROR WORD NUMBER HERE
2054  004604  000000                    TSTNM:  0                       ;TEST NUMBER
2055  004606  000000                    RP4VEC: 0                       ;CONTAINS ADDRESS OF LOCATION
2056                                                                    ;WHERE AN RP04 INTERRUPT IS TO VECTOR TO
2057                                                                    ;THIS MUST BE MOVED INTO 'RPVEC' TO BE
2058                                                                    ;EFFECTIVE.
2059
2060  004610  000000                    OFSTVL: 0                       ;OFFSET VALUE USED IN OFFSET TEST
2061
2062
2063  004612  000024                    SAVERE: .BLKW   20.             ;BLOCK TO SAVE REGISTERS FOR PRETEST
2064                                                                    ;HARDWARE REGISTER SNAPSHOTS - THESE
2065                                                                    ;ARE USUALLY THEN CHANGED TO REFLECT
2066                                                                    ;EXPECTED CONDITIONS AFTER THE TEST
2067  004662  000000                    FINALA: 0                       ;SAVE LOOK AHEAD REGISTER AT END OF OPERATION
2068  004664  000000                    FINACC: 0                       ;SAVE CURRENT CYLINDER REGISTER AT END OF OPERATION
2069
2070
2071                                    ;*TABLE FOR ATTENTION BITS
2072                                    ;*ATTENTION TABLE
2073
2074  004666    001     002     004     ATABLE: .BYTE   1,2,4,10,20,40,100,200
2075  004671    010     020     040
2076  004674    100     200
2077
```

```
2078
2079
2080                                    ;*FLAGS AND INTERNAL PROGRAM CONTROL WORDS
2081
2082  004676  000010                    UNITS:  .BLKW   8.              ;THIS IS FILLED WITH -1
2083  004716  000000                    UNIT:   .WORD   0               ;UNIT UNDER TEST
2084  004720  000000                    NOUNIT: .WORD   0               ;NUMBER OF UNITS PRESENT
2085                                                                    ;USED TO KEEP TRACK OF UNIT UNDER TEST
2086  004722  000000                    NUNIT:  .WORD   0               ;USED TO DETERMIN IF THERE ARE MORE
2087                                                                    ;THAN ONE UNIT
2088  004724  000000                    NOPUSH: 0                       ;ALL ONES INDICATE NONE OF THE OPERATOR
2089                                                                    ;INTERVENTION TESTS WILL BE PERFORMED
2090  004726  000000                    SELECT: .WORD   0               ;ALL ONES INDICATE UNIT TO BE SELECTED
2091  004730  000000                    UNITSL: .WORD   0               ;UNIT NO. SELECTED
2092  004732  000000                    UBUSB:  0                       ;IF ZERO UNIBUS PRESENT
2093                                                                    ;IF ONES NO UNIBUS B
2094  004734  000000                    ERFLG$: 0                       ;ERROR FLAG
2095  004736  000000                    FIRST:  0                       ;IF ZERO WILL TYPE HEADER
2096                                                                    ;IF ONES WILL NOT TYPE HEADER
2097
2098  004740  000000                    ATTENT: 0                       ;ATTENTION BIT FOR PRESENT UNIT
2099  004742  000000                    TOTALAT:0                       ;TOTAL ATTENTION BITS
2100
2101  004744  000000                    RP06:   0                       ;RP06 DEVICE TYPE FLAG LOCATION
2102  004746  000000                    RP05:   0                       ;MEMOREX RP04 DEVICE TYPE FLAG
2103  004750  000000                    RH70:   0                       ;IF = 1, PROGRAM IS RUNNING ON RWP04 SYSTEM
2104                                                                    ;IF = 0, PROGRAM IS RUNNING ON RJP04
2105  004752  000000                    INUNIT: 0                       ;INITIAL UNIT NO. - USED DURING
2106                                                                    ;CHECKING ALL ADDRESS PLUG ADDRESSES
2107
2108
2109
2110
2111
2112  004754  000000                    TMP0:   .WORD   0               ;TEMP STORAGE
2113  004756  000000                    TMP1:   .WORD   0
2114  004760  000000                    TMP4:   .WORD   0               ;TEMP STORAGE
```

```
                                        .SBTTL
                                        .SBTTL  **DIAGNOSTIC CODE**
                                        .SBTTL


                                                .SBTTL  SETUP TESTS


004762  012737  177777  004724  BEGIN1: MOV     #-1,@#NOPUSH    ;JUMP OVER OPERATOR REQUIRED TESTS
004770  005037  004726                  CLR     @#SELECT        ;DO NOT SELECT UNIT
004774  000412                          BR      START
004776  012737  177777  004726  BEGIN2: MOV     #-1,@#SELECT    ;SELECT UNIT
005004  005037  004724                  CLR     @#NOPUSH        ;DO NOT JUMP OVER ANY TEST
005010  000404                          BR      START
005012  005037  004726          BEGIN:  CLR     @#SELECT        ;DO NOT SELECT UNIT
005016  005037  004724                  CLR     @#NOPUSH        ;DO NOT JUMP OVER ANY OPERATOR
                                                                ;INTERVENTION TESTS - NORMAL RUN

005022                          START:
005022  000005                          RESET
                                        .SBTTL  INITIALIZE THE COMMON TAGS
                                        ;;CLEAR THE COMMON TAGS ($CMTAG) AREA
005024  012706  001100                  MOV     #$CMTAG,R6      ;;FIRST LOCATION TO BE CLEARED
005030  005026                          CLR     (R6)+           ;;CLEAR MEMORY LOCATION
005032  022706  001140                  CMP     #SWR,R6 ;;DONE?
005036  001374                          BNE     .-6             ;;LOOP BACK IF NO
005040  012706  001000                  MOV     #STACK,SP       ;;SETUP THE STACK POINTER

                                ;;INITIALIZE A FEW VECTORS
005044  012737  044776  000020          MOV     #$SCOPE,@#IOTVEC ;;IOT VECTOR FOR SCOPE ROUTINE
005052  012737  000340  000022          MOV     #340,@#IOTVEC+2 ;;LEVEL 7
005060  012737  047174  000030          MOV     #$ERROR,@#EMTVEC ;;EMT VECTOR FOR ERROR ROUTINE
005066  012737  000340  000032          MOV     #340,@#EMTVEC+2 ;;LEVEL 7
005074  012737  050720  000034          MOV     #$TRAP,@#TRAPVEC ;;TRAP VECTOR FOR TRAP CALLS
005102  012737  000340  000036          MOV     #340,@#TRAPVEC+2;LEVEL 7
005110  012737  051004  000024          MOV     #$PWRDN,@#PWRVEC ;;POWER FAILURE VECTOR
005116  012737  000340  000026          MOV     #340,@#PWRVEC+2 ;;LEVEL 7
005124  005037  001212                  CLR     $TIMES          ;;INITIALIZE NUMBER OF ITERATIONS
005130  005037  001214                  CLR     $ESCAPE         ;;CLEAR THE ESCAPE ON ERROR ADDRESS
005134  112737  000001  001115          MOVB    #1,$ERMAX       ;;ALLOW ONE ERROR PER TEST
005142  012737  005142  001106          MOV     #.,$LPADR       ;;INITIALIZE THE LOOP ADDRESS FOR SCOPE
005150  012737  005150  001110          MOV     #.,$LPERR       ;;SETUP THE ERROR LOOP ADDRESS
                                ;;SIZE FOR A HARDWARE SWITCH REGISTER. IF NOT FOUND OR IT IS
                                ;;EQUAL TO A "-1", SETUP FOR A SOFTWARE SWITCH REGISTER.
005156  013746  000004                  MOV     @#ERRVEC,-(SP)  ;;SAVE ERROR VECTOR
005162  012737  005216  000004          MOV     #64$,@#ERRVEC   ;;SET UP ERROR VECTOR
005170  012737  177570  001140          MOV     #DSWR,SWR       ;;SETUP FOR A HARDWARE SWICH REGISTER
005176  012737  177570  001142          MOV     #DDISP,DISPLAY  ;;AND A HARDWARE DISPLAY REGISTER
005204  022777  177777  173726          CMP     #-1,@SWR        ;;TRY TO REFERENCE HARDWARE SWR
005212  001012                          BNE     66$             ;;BRANCH IF NO TIMEOUT TRAP OCCURRED
                                                                ;;AND  THE HARDWARE SWR IS NOT = -1
005214  000403                          BR      65$             ;;BRANCH IF NO TIMEOUT
005216  012716  005224          64$:    MOV     #65$,(SP)       ;;SET UP FOR TRAP RETURN
005222  000002                          RTI
005224  012737  000176  001140  65$:    MOV     #SWREG,SWR      ;;POINT TO SOFTWARE SWR
005232  012737  000174  001142          MOV     #DISPREG,DISPLAY
005240  012637  000004          66$:    MOV     (SP)+,@#ERRVEC  ;;RESTORE ERROR VECTOR
```

```
2171
2172
2173
2174
2175  005244  012737  000000  177776          MOV     #0,PS             ;SET PROCESSOR STATUS TO 0
2176  005252  012737  000200  000036          MOV     #200,@#TRAPVEC+2  ;TRAP PRIORITY = 4
2177  005260  013700  002266                  MOV     @#RPVEC,R0        ;GET RP VECTOR ADDRESS
2178  005264  012720  044734                  MOV     #RPVECT,(R0)+     ;THIS IS FOR UNTIMELY INTERRUPTS
2179  005270  012710  000340                  MOV     #340,(R0)         ;DRIVE INTERRUPT SERVICE ROUTINE
2180                                                                     ;PRIORITY = 7
2181
2182  005274  004737  045734                  JSR     PC,@#$TKINT       ;INITIALIZE THE TTY KEYBOARD
2183  005300  005737  004736                  TST     @#FIRST           ;IS THIS FIRST TIME ROUND ?
2184  005304  001001                          BNE     1$                ;DON'T GIVE HEADER IF NOT
2185  005306  000402                          BR      2$                ;HEADER 1ST TIME THROUGH
2186  005310  000137  006112          1$:     JMP     @#SND1            ;NO HEADER
2187
2188  005314                          2$:
2189  005314  104401  005322                  TYPE    ,68$              ;;TYPE ASCIZ STRING
2190  005320  000435                          BR      67$               ;;GET OVER THE ASCIZ
2191                                  ;;68$:  .ASCIZ  <15><12>?RP04/5/6 FUNCTIONAL CONTROLLER TEST - PART I  - CZRJI-D?
2192  005414                          67$:
2193  005414  104401  005422                  TYPE    ,70$              ;;TYPE ASCIZ STRING
2194  005420  000431                          BR      69$               ;;GET OVER THE ASCIZ
2195                                  ;;70$:  .ASCIZ  <15><12>/ALL DCL'S MUST BE LOCKED ON THE CORRECT PORT/<15><12>
2196  005504                          69$:
2197
2198  005504  104401  005512                  TYPE    ,72$              ;;TYPE ASCIZ STRING
2199  005510  000424                          BR      71$               ;;GET OVER THE ASCIZ
2200                                  ;;72$:  .ASCIZ  <15><12>/PROGRAMMABLE DRIVES WILL NOT BE USED/
2201  005562                          71$:
2202  005562  104401  005570                  TYPE    ,74$              ;;TYPE ASCIZ STRING
2203  005566  000433                          BR      73$               ;;GET OVER THE ASCIZ
2204                                  ;;74$:  .ASCIZ  <15><12>/IF CHANGE IS REQUIRED ON PORT SWITCH, THEN A CYCLE/
2205  005656                          73$:
2206  005656  104401  005664                  TYPE    ,76$              ;;TYPE ASCIZ STRING
2207  005662  000435                          BR      75$               ;;GET OVER THE ASCIZ
2208                                  ;;76$:  .ASCIZ  <15><12>/UP SEQUENCE IS REQ FOR STROBING THE PORT SELECT FLOP/<15><12>
2209  005756                          75$:
2210
2211  005756  104401  005764                  TYPE    ,78$              ;;TYPE ASCIZ STRING
2212  005762  000426                          BR      77$               ;;GET OVER THE ASCIZ
2213                                  ;;78$:  .ASCIZ  <15><12>/ALL DCL'S NOT UNDER TEST MUST BE SWITCHED/
2214  006040                          77$:
2215  006040  104401  006046                  TYPE    ,80$              ;;TYPE ASCIZ STRING
2216  006044  000422                          BR      79$               ;;GET OVER THE ASCIZ
2217                                  ;;80$:  .ASCIZ  <15><12>/OFF, OR LOCKED ON THE OTHER PORT/
2218  006112                          79$:
2219
2220  006112  012737  177777  004736  SND1:   MOV     #-1,@#FIRST       ;NEXT TIME DO NOT GIVE HEADER
2221
2222                                  .SBTTL  GET VALUE FOR SOFTWARE SWITCH REGISTER
2223  006120  005737  000042                  TST     @#42              ;;ARE WE RUNNING UNDER XXDP/ACT?
2224  006124  001006                          BNE     64$               ;;BRANCH IF YES
2225  006126  023727  001140  000176          CMP     SWR,#SWREG        ;;SOFTWARE SWITCH REG SELECTED?
2226  006134  001005                          BNE     65$               ;;BRANCH IF NO
```

```
                                        GTSWR                   ;;GET SOFT-SWR SETTINGS
                                        BR      65$
006142  112737  000001  001134  64$:    MOVB    #1,$AUTOB       ;;SET AUTO-MODE INDICATOR
006150                                  65$:

006150  032777  010000  172762  RH70CK: BIT     #SW12,@SWR      ;LOOK TO SEE IF USING RH70
006156  001403                          BEQ     3$              ;IF SW12 = 0, SKIP NEXT
006160  012737  000001  004750          MOV     #1,@#RH70       ;IF SW12 = 1, CU IS AN RH70
006166                                  3$:
                                        ;;**************************************************************************
                                        ;*IS THERE A P-CLOCK (KW11-P) ON THE SYSTEM
                                        ;*IF SO MAKE 'WAT" TRAPS GO TO 'WAIT.P'
                                        ;*IF SO MAKE RP04 INTERRUPTS GO TO 'TIME 1'
                                        ;*IF NOT MAKE 'WAT' TRAPS GO TO 'WAIT.T'
                                        ;*IF NOT MAKE RP04 INTERRUPTS GO TO 'TIME 2'

                                        ;*THE NEXT LINE IS TO BE ADDED LATER
                                        ;*AND THE JUMP AND NOP REMOVED
                                        ;*FOR NOW NO CLOCK WILL BE USED

                                        ;       MOV     @#1$,@#ERRVEC   ;SET TIME-OUT VECTOR
                                        ;
                                        ;       JMP     @#1$            ;DO NOT USE CLOCK
                                        ;       NOP
                                        ;       TST     @#PCLCSR        ;REFERENCE P-CLOCK STATUS REGISTER
                                        ;                               ;ADDRESS = 172540
                                        ;       MOV     #WAIT.P,@#$TRPAD+20 ;THERE IS A P-CLOCK
                                        ;       MOV     #TIME1,@#RP4VEC ;THERE IS A P CLOCK SO
                                        ;                               ;VECTOR TO TIME1
                                        ;       BR      2$
                                        ;1$:    MOV     #WAIT.T,@#$TRPAD+20 ;THERE IS NO P-CLOCK
                                        ;;**************************************************************************


006166  012737  041736  004606          MOV     #TIME2,@#RP4VEC ;RP04/5/6 INTERRUPTS GO TO 'TIME 2'
006174  012737  177777  047342  2$:     MOV     #-1,@#PRITEM    ;CLEAR PREVIOUS ITEM NUMBER


006202  005737  004726                  TST     @#SELECT        ;WAS IT A 200 START
006206  001442                          BEQ     TST1    ;      BRANCH IF STARTING FROM 200
006210  104401  006216                  TYPE    ,65$            ;;TYPE ASCIZ STRING
006214  000424                          BR      64$             ;;GET OVER THE ASCIZ
                                ;;65$:  .ASCIZ  <15><12>/SELECT UNIT NUMBER TO BE TESTED ?  /<15><12>
006266                          64$:
006266  104412                          RDOCT
006270  042716  177770                  BIC     #177770,(SP)    ;ONLY KEEP LAST 3 BITS
006274  011637  004716                  MOV     (SP),@#UNIT     ;SAVE UNIT TO BE TESTED
006300  012637  004730                  MOV     (SP)+,@#UNITSL  ;SAVE UNIT TO BE TESTED


006304  001403                          BEQ     TST1    ;      BRANCH IF STARTING FROM 200
006306  013737  004730  004716          MOV     @#UNITSL,@#UNIT ;SET UNIT NUMBER
```

```
2283
```

```
2284
2285                                    ;;*********************************************************************
2286                                    ;*TEST 1         REFERENCE EACH REGISTER
2287
2288                                    ;*              REFERENCE EACH REGISTER BY A MOVE INSTRUCTION
2289
2290                                    ;;*********************************************************************
2291  006314  000004                    TST1:   SCOPE
2292  006316  012737  000001  001212            MOV     #1,$TIMES       ;;DO 1 ITERATION
2293  006324  012737  000001  004604            MOV     #2-1,@#TSTNM    ;THIS SAVES TEST NUMBER
2294  006332  012706  001000                    MOV     #STACK, SP      ;SET UP STACK POINTER
2295  006336  012737  047204  000030            MOV     #REGSA1,@#EMTVEC;ERROR VECTOR SO THAT
2296                                                                    ;NO REGISTERS ARE SAVED
2297  006344  012737  006372  000004            MOV     #2$,@#ERRVEC    ;SET UP FOR BUS TIMEOUT
2298
2299  006352  012700  000024                    MOV     #24,R0          ;THERE ARE 24 REG TO TEST
2300  006356  012701  002270                    MOV     #RHDB,R1        ;R1 NOW HAS ADDR OF ADDR OF FIRST REG.
2301  006362  013102                    1$:     MOV     @(R1)+,R2       ;READ HARDWARE REG.
2302  006364  005300                            DEC     R0              ;COUNT DOWN
2303  006366  001375                            BNE     1$              ;BRANCH IF 24 NOT DONE
2304  006370  000454                            BR      3$              ;BRANCH IF 24 DONE
2305  006372  012737  000006  000004   2$:     MOV     #ERRVEC+2,@#ERRVEC ;RESTORE TRAP CATCHER
2306  006400  022626                            CMP     (SP)+,(SP)+     ;CLEAN STACK
2307  006402  016137  177776  001200            MOV     -2(R1),$TMP1    ;STORE FAILING REG ADDR
2308  006410  104007                            ERROR   7               ;REGISTER NON EXISTANT
2309  006412  032777  020000  172520            BIT     #SW13,@SWR      ;INHIBIT ERROR PRINTOUT ?
2310  006420  001036                            BNE     4$              ;BRANCH IF YES
2311
2312  006422  104401  006430                    TYPE    ,65$            ;;TYPE ASCIZ STRING
2313  006426  000427                            BR      64$             ;;GET OVER THE ASCIZ
2314                                    ;;65$:  .ASCIZ  <15><12>/TO CHANGE BASE ADDRESS RESTART AT ADDRESS /
2315  006506                            64$:
2316
2317  006506  012746  000204                    MOV     #ADDMOD,-(SP)   ;GET READY TO TYPE STARTING ADDRESS
2318                                                                    ;OF ''CHANGE OF BASE ADDRESS'' ROUTINE
2319  006512  104402                            TYPOC
2320  006514  000000                            HALT                    ;STOP TO FORCE THE CORRECT RESTART
2321  006516  000137  041052            4$:     JMP     @#$EOP          ;GO TO END OF PROGRAM ------------->
2322
2323  006522  012737  006602  000004   3$:     MOV     #TRP,@#4        ;INITIALIZE VECTOR
2324  006530  005037  004750                    CLR     RH70            ;INIT RH INDICATOR ++ C.W
2325  006534  005777  173600                    TST     @RHBAE          ;ADDRESS RPBAE(RH11/RH70?)
2326  006540  005237  004750                    INC     RH70            ;FOUND AN RH70-SET MASK
2327  006544  104401  006552                    TYPE    ,67$            ;;TYPE ASCIZ STRING
2328  006550  000413                            BR      66$             ;;GET OVER THE ASCIZ
2329                                    ;;67$:  .ASCIZ  <15><12>/RH70 CONTROLLER   /
2330  006600                            66$:
2331  006600  000420                            BR      RTN             ;AND GET OUT
2332  006602  005726                    TRP:    TST     (SP)+
2333  006604  005726                            TST     (SP)+           ;RESTORE THE STACK
2334  006606  104401  006614                    TYPE    ,65$            ;;TYPE ASCIZ STRING
2335  006612  000413                            BR      64$             ;;GET OVER THE ASCIZ
2336                                    ;;65$:  .ASCIZ  <15><12>/RH11 CONTROLLER   /
2337  006642                            64$:
2338  006642  012737  047174  000030   RTN:    MOV     #$ERROR,@#EMTVEC;RESTORE ERROR VECTOR
2339                                                                    ;SO THAT REGISTERS ARE SAVED
```

```
 2340  006650  012737  000006  000004          MOV     #ERRVEC+2,@#ERRVEC ;RESTORE TRAP CATCHER
 2341
```

```
2342
2343
2344                                    ;;*******************************************************************
2345                                    ;*TEST 2        PARTIAL TEST OF RHAS FOR UNIT NUMBERS PRESENT
2346
2347                                    ;*      CHECK THAT RHAS CAN BE CLEARED BY MOVING ALL ONES INTO IT
2348
2349                                    ;;*******************************************************************
2350  006656  000004                   TST2:   SCOPE
2351  006660  012737  000001  001212           MOV     #1,$TIMES       ;;DO 1 ITERATION
2352  006666  012706  001000                   MOV     #STACK,SP       ;SET STACK POINTER
2353  006672  012737  000002  004604           MOV     #3-1,@#TSTNM    ;THIS SAVES TEST NUMBER
2354
2355  006700  013701  002316                   MOV     @#RHAS,R1       ;R1 HAS ADDRESS OF RHAS
2356  006704  012711  177777                   MOV     #-1,@R1         ;THIS WRITES ALL ONES INTO RHAS
2357  006710  105711                           TSTB    @R1             ;TEST RHAS FOR ALL 0'S
2358  006712  001407                           BEQ     TST3    ;       BRANCH IF GOOD
2359  006714  011137  001126                   MOV     @R1,@#$BDDAT    ;BAD DATA
2360  006720  005037  001124                   CLR     @#$GDDAT        ;GOOD DATA
2361  006724  010137  004600                   MOV     R1,@#REGADR     ;FAILING REG. (RHAS)
2362
2363  006730  104005                           ERROR   5               ;RHAS DOES NOT CLEAR BY WRITING
2364                                                                   ;ALL ONES INTO IT
2365
```

```
2366
2367                                    ;;*****************************************************************
2368                                    ;*TEST 3        TEST FOR DRIVES PRESENT USING RHAS AND RHCS2
2369
2370                                    ;*      THE NUMBER OF DRIVES PRESENT IS FOUND
2371                                    ;*      BY MOVING ALL ONES INTO RHER1 WITH UNIT NUMBER
2372                                    ;*      IN RHCS2 INCREMENTED FROM ZERO TO SEVEN
2373
2374                                    ;*      THE BITS SET IN RHAS SHOULD GIVE DRIVES PRESENT
2375
2376                                    ;*      THE DRIVE TYPE IS CHECKED TO BE AN RP04/RP06 AND THEN
2377                                    ;*      UNITS PRESENT ARE STORED IN A TABLE CALLED 'UNITS'
2378
2379                                    ;;*****************************************************************
2380  006732  000004                   TST3:   SCOPE
2381  006734  012737  000001  001212           MOV     #1,$TIMES       ;;DO 1 ITERATION
2382  006742  012737  000003  004604           MOV     #4-1,@#TSTNM    ;THIS SAVES TEST NUMBER
2383  006750  000005                           RESET                   ;START WITH AN INIT
2384  006752  004737  045734                   JSR     PC,@#$TKINT     ;INITIALIZE TTY KEYBOARD
2385
2386  006756  032777  020000  172154           BIT     #SW13,@SWR      ;INHIBIT ERROR TYPEOUT?
2387  006764  001026                           BNE     4$              ;BRANCH IF YES
2388  006766  104401  006774                   TYPE    ,65$            ;;TYPE ASCIZ STRING
2389  006772  000423                           BR      64$             ;;GET OVER THE ASCIZ
2390                                    ;;65$:  .ASCIZ  <15><12><15><12>/LOOKING AT RHAS - DRIVES PRESENT /
2391  007042                            64$:
2392  007042  013701  002316           4$:     MOV     @#RHAS,R1       ;R1 HAS ADDR. OF RHAS
2393  007046  013702  002276                   MOV     @#RHCS2,R2      ;R2 HAS ADDR. OF RHCS2
2394  007052  005012                           CLR     @R2             ;CLEAR RHCS2
2395  007054  012700  000010                   MOV     #8.,R0          ;COUNT
2396  007060  013704  002302                   MOV     @#RHER1,R4      ;R4 HAS ADDR. OF RHER1
2397  007064  012714  177777           1$:     MOV     #-1,@R4         ;MOVE ERRORS INTO RHER1
2398  007070  005212                           INC     @R2             ;INCREMENT UNIT NO.
2399  007072  005300                           DEC     R0              ;COUNT DOWN DRIVE COUNTER
2400  007074  001373                           BNE     1$              ;DO NEXT UNIT IF 8 NOT DONE
2401
2402  007076  111137  004742                   MOVB    @R1,@#TOTALAT   ;SAVE TOTAL ATTENTION
2403                                                                    ;USED IN DRIVE CLEAR TEST
2404  007102  105037  004743                   CLRB    @#TOTALAT+1     ;CLEAR UPPER BYTE
2405  007106  105711                           TSTB    @R1             ;TEST 'RHAS' FOR ANY DRIVES PRESENT
2406  007110  001402                           BEQ     2$              ;NONE RESPONDING - TYPE THE MESSAGE
2407  007112  000137  007464                   JMP     XE2             ;SOME THERE - GO FILL ''UNITS'' TABLE
2408
2409  007116  032777  020000  172014   2$:     BIT     #SW13,@SWR      ;INHIBIT ERROR TYPE OUT?
2410  007124  001402                           BEQ     3$              ;''NO DRIVES'' MESSAGE IF NOT
2411  007126  000137  010066                   JMP     SELTST          ;CHECK FOR SELECTED UNIT START AND LOAD
2412                                                                    ;''UNITS'' TABLE WITH SELECTED ONE IF SO
2413
2414  007132                            3$:
2415  007132  104401  007140                   TYPE    ,67$            ;;TYPE ASCIZ STRING
2416  007136  000420                           BR      66$             ;;GET OVER THE ASCIZ
2417                                    ;;67$:  .ASCIZ  <15><12>/NO DRIVES PRESENT - RHAS = 0/
2418  007200                            66$:
2419  007200  104401  007206                   TYPE    ,69$            ;;TYPE ASCIZ STRING
2420  007204  000430                           BR      68$             ;;GET OVER THE ASCIZ
2421                                    ;;69$:  .ASCIZ  <15><12>/WRITING ONES INTO RHER1 FOR ALL UNIT NUMBERS/
```

```
2422  007266                          68$:
2423  007266  104401  007274                  TYPE    ,71$            ;;TYPE ASCIZ STRING
2424  007272  000430                  BR      70$             ;;GET OVER THE ASCIZ
2425                          ;;71$:  .ASCIZ  <15><12>/DOES NOT SET ANY BIT IN RHAS SO ABORT PROGRAM/
2426  007354                          70$:
2427  007354  104401  007362                  TYPE    ,73$            ;;TYPE ASCIZ STRING
2428  007360  000437                  BR      72$             ;;GET OVER THE ASCIZ
2429                          ;;73$:  .ASCIZ  <15><12>/TO LOOP ON THIS TEST WO PRINTOUT SET SWITCHES 13, 8, 1, & 0/
2430  007460                          72$:
2431
2432  007460  000137  041052                  JMP     @#$EOP          ;GO OUT------------------------------>
2433
2434
2435
2436                                  ;*SET UP UNITS TABLE
2437
2438  007464                          XE2:
2439  007464  012700  000010          2$:     MOV     #8.,R0          ;COUNTER
2440  007470  012703  004676                  MOV     #UNITS,R3       ;POINTER
2441  007474  012723  177777          3$:     MOV     #-1,(R3)+       ;PRESET BLOCK TO ALL ONES
2442  007500  005300                  DEC     R0              ;COUNT
2443  007502  001374                  BNE     3$              ;BRANCH IF 8 NOT DONE
2444  007504  012703  004676                  MOV     #UNITS,R3       ;POINTER
2445  007510  005005                  CLR     R5              ;INITIALIZE UNIT NO. TO 0
2446  007512  005037  004720                  CLR     @#NOUNIT        ;NO. OF UNITS PRESENT
2447  007516  012700  000010                  MOV     #8.,R0          ;COUNTER
2448  007522  011137  001176                  MOV     @R1,@#$TMP0     ;TEMPORARY STORAGE
2449  007526  006037  001176          4$:     ROR     @#$TMP0         ;SET CARRY IF ONE IN  0 BIT
2450  007532  103135                  BCC     5$              ;CHECK NEXT UNIT IF ONE NOT IN BIT 0
2451
2452  007534  010577  172536                  MOV     R5,@RHCS2       ;INSERT UNIT NUMBER INTO RHCS2 UA BITS
2453  007540  022777  024020  172556          CMP     #24020,@RHDT    ;IS THIS A DUAL PORT RP04 ?
2454  007546  001425                  BEQ     7$              ;TYPE THE UNIT NO. IF YES
2455  007550  022777  020020  172546          CMP     #20020,@RHDT    ;IS THIS A SINGLE PORT RP04 ?
2456  007556  001421                  BEQ     7$              ;TYPE UNIT NO. IF YES
2457
2458                          ;;*****************************************************************
2459  007560  022777  024021  172536          CMP     #24021,@RHDT    ;DUAL PORT RP05 ?
2460  007566  001415                  BEQ     7$              ;TYPE UNIT NO. IF SO
2461  007570  022777  020021  172526          CMP     #20021,@RHDT    ;SINGLE PORT RP05 ?
2462  007576  001411                  BEQ     7$              ;TYPE UNIT NO. IF SO
2463
2464  007600  022777  024022  172516          CMP     #24022,@RHDT    ;IS THIS A DUAL PORT RP06 ?
2465  007606  001405                  BEQ     7$              ;TYPE THE UNIT NO. IF SO
2466  007610  022777  020022  172506          CMP     #20022,@RHDT    ;IS THIS A SINGLE PORT RP06 ?
2467  007616  001401                  BEQ     7$              ;TYPE UNIT NO. IF SO
2468  007620  000414                  BR      9$
2469  007622  032777  001000  172472  7$:     BIT     #BIT09, @RHDS1  ;IS THE DRIVE PROGRAMMABLE?
2470  007630  001001                  BNE     8$              ;BRANCH IF YES
2471  007632  000466                  BR      6$
2472  007634  104401  001223          8$:     TYPE    ,$CRLF
2473  007640  010546                  MOV     R5,     -(SP)
2474  007642  104405                  TYPDS                   ;TYPE THE DRIVE NUMBER
2475  007644  104401  066766                  TYPE    ,NOUSE          ;REPORT THIS DRIVE WILL NOT BE USED(DRIVE MUST BE LOCKED
2476  007650  000466                  BR      5$
2477  007652                          9$:
```

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

                                        ;*NO...IT'S NOT AN RP04/RP05/RP06 DEVICE SO TYPE
                                        ;*OUT THE DEVICE TYPE

007652  104401  007660                    TYPE    ,65$            ;;TYPE ASCIZ STRING
007656  000410                            BR      64$             ;;GET OVER THE ASCIZ
                                ;;65$:  .ASCIZ  <15><12>/UNIT NUMBER /
007700                          64$:
007700  010546                            MOV     R5,-(SP)        ;GET READY TO TYPE UNIT NUMBER
007702  104405                            TYPDS
007704  104401  007712                    TYPE    ,67$            ;;TYPE ASCIZ STRING
007710  000406                            BR      66$             ;;GET OVER THE ASCIZ
                                ;;67$:  .ASCIZ  /,  RHDT = /
007726                          66$:
007726  017746  172372                    MOV     @RHDT,-(SP)     ;GET READY TO TYPE RHDT
007732  104402                            TYPOC
007734  104401  007742                    TYPE    ,69$            ;;TYPE ASCIZ STRING
007740  000422                            BR      68$             ;;GET OVER THE ASCIZ
                                ;;69$:  .ASCIZ  ? - NOT AN RP04/RP05/RP06 DEVICE !!?
010006                          68$:
010006  000407                            BR      5$              ;NO RP04/RP05/RP06 FOUND SO TEST NEXT UNIT

010010  010523                  6$:       MOV     R5,(R3)+        ;LOAD TABLE POSITION AND INCR IT
010012  104401  001223                    TYPE    ,$CRLF
010016  010546                            MOV     R5,-(SP)        ;PUT DRIVE NO. ON STACK
010020  104405                            TYPDS                   ;TYPE DRIVE NO.
010022  005237  004720                    INC     @#NOUNIT        ;INCR THE TOTAL NO. OF UNITS

010026  005205                  5$:       INC     R5              ;'RHCS2' UNIT ADDRESS
010030  005300                            DEC     R0              ;DRIVE COUNTER DOWN ONE
010032  001235                            BNE     4$              ;TEST AND DO NEXT UNIT IF 8 NOT DONE

010034  005737  004720                    TST     @#NOUNIT        ;IF THERE ARE ANY UNITS...
010040  001002                            BNE     10$             ;CONTINUE
010042  000137  041052                    JMP     @#$EOP          ;ELSE GO TO END OF PASS
010046                          10$:

010046  013737  004676  004716            MOV     @#UNITS,@#UNIT  ;SET UNIT NO. TO FIRST ONE FOUND OR 0
010054  013737  004720  004722            MOV     @#NOUNIT,@#NUNIT;SAVE NO. OF UNITS
010062  005337  004722                    DEC     @#NUNIT         ;IF NUNIT = 0 THEN ONLY ONE UNIT
                                                                ;IF NUNIT > 0 THEN MORE THAN ONE UNIT

010066  005737  004726          SELTST:   TST     @#SELECT        ;STARTING ADDRESS 200
010072  001403                            BEQ     TST4    ;      SKIP NEXT IF STARTING FROM 200
010074  013737  004730  004716            MOV     @#UNITSL,@#UNIT ;CHANGE UNIT NUMBER TO SELECTED ONE
```

```
                                        ;;*****************************************************************
                                        ;*TEST 4        TYPE SERIAL NUMBER AND DRIVE TYPE

                                        ;*      SET APPROPRIATE ATTTENTION BIT OF UNIT UNDER TEST IN 'ATTENT'
                                        ;*      TYPE UNIT UNDER TEST

                                        ;*      READ SERIAL NUMBER AND DRIVE TYPE REGISTERS
                                        ;*      TYPE THEM OUT AND PROCEED

                                        ;*      TO LOOP HERE SET SWITCH 8, AND THIS TEST NUMBER ON
                                        ;*      SWITCHES 0 THRU 7 AND RESTART

                                        ;;*****************************************************************
010102  000004                  TST4:   SCOPE
010104  012737  000001  001212          MOV     #1,$TIMES       ;;DO 1 ITERATION
010112  012737  011044  001106          MOV     #1$,$LPADR      ;;SET SCOPE LOOP ADDRESS
010120  012706  001000                  MOV     #STACK,SP       ;RESET STACK
010124  012737  000004  004604          MOV     #4,@#TSTNM      ;SAVE TEST NUMBER

010132  004737  041524                  JSR     PC,@#CLDISK     ;SET R1-RHCS1, R2-RHCS2
                                                                ;R3-RHDS1, R4-RHER1
                                                                ;GIVE RH-11 INITIALIZE
                                                                ;SETUP UNIT NUMBER
010136  005037  004740                  CLR     @#ATTENT        ;CLEAR

010142  005737  004716                  TST     @#UNIT          ;IS "UNIT 0" NEXT IN THE UNITS TABLE ?
010146  001107                          BNE     9$              ;IF NOT TEST THIS UNIT
010150  012700  000041                  MOV     #41,R0          ;IF SO, CHECK THE LOAD MEDIA LOCATION
010154  122710  000011                  CMPB    #11,(R0)        ;WAS IT AN RP04/5/6 ?
010160  001102                          BNE     9$              ;NO - GO AHEAD WITH TESTING UNIT #0

010162  005737  004726                  TST     @#SELECT        ;WAS UNIT #0 SELECTED ?
                                                                ;(IE. WAS IT A 210 START ?)
010166  001006                          BNE     12$             ;IF SO, CHANGE PACKS

                                        ;*INCREMENT THE UNITS TABLE TO NEXT DRIVE (IF ANY)
                                        ;*& DECREMENT THE "NOUNITS" PRESENT (TO BE TESTED)

010170  012700  004676                  MOV     #UNITS,R0       ;LOAD UNITS TABLE POINTER
010174  005720                          TST     (R0)+           ;SELECT THE NEXT UNIT IN THE TABLE
                                                                ;(DOUBLE INCREMENT THE POINTER, R0)
010176  022710  177777                  CMP     #-1,(R0)        ;IS THERE ANOTHER TABLE ENTRY PRESENT ?
010202  001065                          BNE     11$             ;IF SO, USE THE NEXT DRIVE & DEC "NOUNITS"
                                                                ;IF NOT, MUST USE DRIVE #0 & CHANGE PACK
010204                          12$:
010204  104401  010212                  TYPE    ,65$            ;;TYPE ASCIZ STRING
010210  000434                          BR      64$             ;;GET OVER THE ASCIZ
                                ;;65$:  .ASCIZ  <15><12><15><12>/DISMOUNT PACK FROM UNIT #0 AND MOUNT A SCRATCH PACK/
010302                          64$:
010302  104401  010310                  TYPE    ,67$            ;;TYPE ASCIZ STRING
010306  000421                          BR      66$             ;;GET OVER THE ASCIZ
                                ;;67$:  .ASCIZ  <15><12>/PRESS CONTINUE WHEN FINISHED/<15><12>
010352                          66$:
```

```
2581  010352  000000                        HALT
2582  010354  000404                        BR      9$              ;CONTINUE, USING SCRATCH PACK ON UNIT #0
2583
2584  010356  011037  004716          11$:  MOV     (R0),@#UNIT     ;SET UP TO BE THE UNIT UNDER TEST
2585  010362  005337  004720                DEC     @#NOUNITS       ;DECREMENT BECAUSE UNIT #0 WON'T BE TESTED
2586
2587  010366  013700  004716          9$:   MOV     @#UNIT,R0       ;R0 CONTAINS UNIT UNDER TEST
2588
2589
2590
2591                                  ;;**********************************************************************
2592  010372  005037  004744                CLR     @#RP06          ;CLEAR RP06 DEVICE TYPE FLAG
2593  010376  010077  171674                MOV     R0,@RHCS2       ;SET UP UNIT ADDRESSING
2594  010402  022777  024022  171714        CMP     #24022,@RHDT    ;IS IT A DUAL PORT RP06 ?
2595  010410  001405                        BEQ     2$              ;YES...SET THE FLAG
2596  010412  022777  020022  171704        CMP     #20022,@RHDT    ;IS IT A SINGLE PORT RP06 ?
2597  010420  001401                        BEQ     2$              ;YES...SET FLAG
2598  010422  000404                        BR      3$              ;DON'T SET FLAG - CHECK FOR RP04
2599  010424  012737  177777  004744  2$:   MOV     #-1,@#RP06      ;SET THE FLAG
2600  010432  000416                        BR      8$              ;DON'T CHECK FOR RP04, IT WAS RP06
2601
2602  010434  005037  004746          3$:   CLR     @#RP05          ;CLEAR MEMOREX RP04 DEVICE FLAG
2603  010440  022777  024021  171656        CMP     #24021,@RHDT    ;IS IT A DUAL PORT MEMOREX RP04 ?
2604  010446  001405                        BEQ     7$              ;YES..SET THE FLAG FOR ADDR PLUG TESTS
2605  010450  022777  020021  171646        CMP     #20021,@RHDT    ;IS IT A SINGLE PORT MEMOREX RP04 ?
2606  010456  001401                        BEQ     7$              ;YES..SET THE FLAG FOR ADDR PLUG TESTS
2607  010460  000403                        BR      8$              ;DON'T SET FLAG - NOT MEMOREX DRIVE
2608  010462  012737  177777  004746  7$:   MOV     #-1,@#RP05      ;SET THE FLAG
2609  010470                          8$:                           ;ASSUME THE NEXT UNIT IS AN RP04
2610                                  ;;**********************************************************************
2611
2612
2613
2614  010470  116037  004666  004740        MOVB    ATABLE(R0),@#ATTENT ;SET APPROPRIATE ATTENTION BIT
2615  010476  104401  010504                TYPE    ,69$            ;;TYPE ASCIZ STRING
2616  010502  000414                        BR      68$             ;;GET OVER THE ASCIZ
2617                                  ;;69$:  .ASCIZ  <15><12>/TESTING DRIVE NUMBER/
2618  010534                          68$:
2619  010534  013746  004716                MOV     @#UNIT,-(SP)    ;UNIT NO. TO STACK
2620  010540  104405                        TYPDS                   ;TYPE  DRIVE NO.
2621  010542  104401  010550                TYPE    ,71$            ;;TYPE ASCIZ STRING
2622  010546  000410                        BR      70$             ;;GET OVER THE ASCIZ
2623                                  ;;71$:  .ASCIZ  <15><12>/SERIAL NO. = /
2624  010570                          70$:
2625  010570  017746  171532                MOV     @RHSN,-(SP)     ;;SAVE @RHSN FOR TYPEOUT
2626  010574  104402                        TYPOC                   ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
2627  010576  104401  010604                TYPE    ,73$            ;;TYPE ASCIZ STRING
2628  010602  000410                        BR      72$             ;;GET OVER THE ASCIZ
2629                                  ;;73$:  .ASCIZ  <15><12>/DRIVE TYPE = /
2630  010624                          72$:
2631  010624  017746  171474                MOV     @RHDT,-(SP)     ;;SAVE @RHDT FOR TYPEOUT
2632  010630  104402                        TYPOC                   ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
2633
```

```
                                        ;TYPE OUT THE DRIVE TYPE IN ASCII


                                ;;**************************************************************
010632  022777  024020  171464          CMP     #24020,@RHDT    ;DUAL PORT RP04 ?
010640  001425                          BEQ     4$              ;TYPE ASCII MSG OUT
010642  022777  020020  171454          CMP     #20020,@RHDT    ;SINGLE PORT RP04 ?
010650  001421                          BEQ     4$              ;TYPE THE MESSAGE

010652  022777  024021  171444          CMP     #24021,@RHDT    ;DUAL PORT RP05 ?
010660  001453                          BEQ     6$              ;TYPE THE MESSAGE
010662  022777  020021  171434          CMP     #20021,@RHDT    ;SINGLE PORT RP05 ?
010670  001447                          BEQ     6$              ;TYPE THE MESSAGE

010672  022777  024022  171424          CMP     #24022,@RHDT    ;DUAL PORT RP06 ?
010700  001424                          BEQ     5$              ;TYPE THE MESSAGE
010702  022777  020022  171414          CMP     #20022,@RHDT    ;SINGLE PORT RP06 ?
010710  001420                          BEQ     5$              ;TYPE THE MESSAGE
010712  000454                          BR      1$              ;DRIVE IS NOT AN RP04/RP05/RP06
                                                                ;DON'T TYPE ASCII MESSAGE OUT

                                                                ;-SHOULD NEVER HAPPEN AT THIS POINT
                                                                ;UNLESS DRIVE GOT SICK WHILE TESTING
                                                                ;WAS IN PROGRESS

010714                          4$:
010714  104401  010722                        TYPE    ,75$            ;;TYPE ASCIZ STRING
010720  000413                          BR      74$             ;;GET OVER THE ASCIZ
                                ;;75$:  .ASCIZ  <15><12>/DRIVE IS AN RP04/<15><12>
010750                          74$:
010750  000435                          BR      1$              ;SKIP NEXT MESSAGE
010752                          5$:
010752  104401  010760                        TYPE    ,77$            ;;TYPE ASCIZ STRING
010756  000413                          BR      76$             ;;GET OVER THE ASCIZ
                                ;;77$:  .ASCIZ  <15><12>/DRIVE IS AN RP06/<15><12>
011006                          76$:
011006  000416                          BR      1$              ;SKIP NEXT MESSAGE
011010                          6$:
011010  104401  011016                        TYPE    ,79$            ;;TYPE ASCIZ STRING
011014  000413                          BR      78$             ;;GET OVER THE ASCIZ
                                ;;79$:  .ASCIZ  <15><12>/DRIVE IS AN RP05/<15><12>
011044                          78$:
                                ;;**************************************************************


011044  005777  171256          1$:     TST     @RHSN           ;READ SERIAL NO. AND DRIVE TYPE
011050  005777  171250                  TST     @RHDT           ;THESE TWO ARE TO HELP SCOPE LOOPS
011054  017737  171246  002410          MOV     @RHSN,@#SN      ;SAVE TO CHECK IF DRIVE CLEAR CLEARS ANY BITS
011062  017737  171236  002406          MOV     @RHDT,@#DT      ;SAVE TO CHECK IF DRIVE CLEAR CLEARS ANY BITS
```

```
                                        ;;*****************************************************************
                                        ;*TEST 5        CHECK MOL TO BE HIGH

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

                                        ;;*****************************************************************
011070  000004                          TST5:   SCOPE
011072  012737  000005  004604                  MOV     #6-1,@#TSTNM    ;THIS SAVES TEST NUMBER
011100  004737  041524                          JSR     PC,@#CLDISK     ;GIVE INITIALIZE
011104  032713  010000                          BIT     #MOL,@R3        ;CHECK MOL IN RHDS1
011110  001151                              BNE     TST6    ;       BRANCH IF MOL HIGH
011112  104401  011120                          TYPE    ,65$            ;;TYPE ASCIZ STRING
011116  000420                              BR      64$             ;;GET OVER THE ASCIZ
                                        ;;65$:  .ASCIZ  <15><12>/DRIVE IS OFFLINE - MOL IS LOW/
011160                                  64$:
011160  104401  011166                          TYPE    ,67$            ;;TYPE ASCIZ STRING
011164  000424                              BR      66$             ;;GET OVER THE ASCIZ
                                        ;;67$:  .ASCIZ  <15><12>/HIT START ON DRIVE TO GET IT ON LINE/
011236                                  66$:
011236  104401  011244                          TYPE    ,69$            ;;TYPE ASCIZ STRING
011242  000431                              BR      68$             ;;GET OVER THE ASCIZ
                                        ;;69$:  .ASCIZ  <15><12>/PROGRAM WILL HANG TESTING MOL TILL MOL IS HIGH/
011326                                  68$:
011326  032713  010000                  1$:     BIT     #MOL,@R3        ;CHECK MOL IN RHDS1
011332  001775                              BEQ     1$              ;BRANCH IF MOL IS HIGH
011334  104401  011342                          TYPE    ,71$            ;;TYPE ASCIZ STRING
011340  000435                              BR      70$             ;;GET OVER THE ASCIZ
                                        ;;71$:  .ASCIZ  <15><12>/GOOD - MOL IS NOW HIGH. PROGRAM WILL NOW BE EXECUTED/<15><12>
011434                                  70$:
```

```
                                ;;*****************************************************************
                                ;*TEST 6        PROGRAM INTERRUPT
                                ;
                                ;*      PROGRAM INTERRUPT IS TESTED BY SETTING RDY AND IE
                                ;*      IN RHCS1 AT THE SAME TIME
                                ;*      THIS SHOULD INTERRUPT THROUGH LOCATION 254
                                ;*      THE PROCESSOR PRIORITY IS SET TO 4
                                ;;*****************************************************************
011434  000004                  TST6:   SCOPE
                                
011436  012737  000006  004604          MOV     #7-1,@#TSTNM    ;THIS SAVES TEST NUMBER
011444  012706  001000                  MOV     #STACK,SP       ;RESET STACK
                                
011450  004737  041524                  JSR     PC,@#CLDISK     ;SET R1-RHCS1, R2-RHCS2
                                                                ;R3-RHDS1, R4-RHER1
                                                                ;GIVE RH-11 INITIALIZE
                                                                ;SETUP UNIT NUMBER
                                
011454  013700  002266                  MOV     @#RPVEC,R0      ;GET RP VECTOR ADDRESS
011460  012720  011526                  MOV     #RPTRP1,(R0)+   ;THIS IS FOR TIMELY INTERRUPTS
011464  012710  000340                  MOV     #340,(R0)       ;RP04 INTERRUPT SERVICE ROUTINE
                                                                ;PRIORITY = 7
011470  012737  000200  177776          MOV     #200,PS         ;SET PROCESSOR PRIORITY @ 4 (DISK @ 5)
011476  012711  000300                  MOV     #RDY!IE,@R1     ;RDY, IE IN RHSC1 SHOULD CAUSE INTERRUPT
                                
011502  013737  042222  001200          MOV     @#TIMCNT,@#$TMP1;COUNTER
011510  005337  001200          1$:     DEC     @#$TMP1         ;WAIT FOR INTERRUPT
011514  001375                  BNE     1$                      ;BRANCH IF NOT ZERO
                                                                ;BEFORE THIS IS ZERO INTERRUPT SHOULD OCCUR
011516  104065                  ERROR   65                      ;INTERRUPT DID NOT OCCUR
011520  012712  000040          MOV     #40,@R2                 ;CLEAR CONTROLLER VIA CS2 CLR BIT
011524  000407                  BR      TST7    ;              BRANCH TO NEXT TEST ----------------)
                                
011526  022626          RPTRP1: CMP     (SP)+,(SP)+             ;RESTORE STACK
011530  022711  004200          CMP     #DVA!RDY,@R1            ;IE SHOULD BE LOW AS RUPT OCCURRED
011534  001403                  BEQ     TST7    ;              CONTINUE IF GOOD ------------------)
011536  104065                  ERROR   65                      ;INTERRUPT OCCURED BUT
                                                                ;IE FAILED TO RESET
011540  012712  000040          MOV     #40,@R2                 ;CLEAR CONTROLLER VIA CS2 CLR BIT
```

```
                                ;;*****************************************************************
                                ;*TEST 7         INTERRUPT AT PROCESSOR AND DISK PRIORITY SAME

                                ;*      PROCESSOR PRIORITY IS SET AT 5 (SAME AS THE DISK)
                                ;*      IE AND RDY IS SET. THIS SHOULD NOT ALLOW INTERRUPT
                                ;;*****************************************************************
011544  000004                  TST7:   SCOPE

011546  012737  000007  004604          MOV     #10-1,@#TSTNM   ;THIS SAVES TEST NUMBER
011554  012706  001000                  MOV     #STACK,SP       ;RESET STACK

011560  004737  041524                  JSR     PC,@#CLDISK     ;SET R1-RHCS1, R2-RHCS2
                                                                ;R3-RHDS1, R4-RHER1
                                                                ;GIVE RH-11 INITIALIZE
                                                                ;SETUP UNIT NUMBER

011564  013700  002266                  MOV     @#RPVEC,R0      ;GET RP VECTOR ADDRESS
011570  012720  011634                  MOV     #RPTRP2,(R0)+   ;THIS IS FOR UNTIMELY INTERRUPTS
011574  012710  000340                  MOV     #340,(R0)       ;RP04 INTERRUPT SERVICE ROUTINE
                                                                ;PRIORITY = 7
011600  012737  000240  177776          MOV     #240,PS         ;SET PROCESSOR PRIORITY @ 5
011606  012711  000300                  MOV     #RDY!IE,@R1     ;RDY, IE IN RHSC1 WHOULD CAUSE INTERRUPT

011612  013737  042222  001200          MOV     @#TIMCNT,@#$TMP1;COUNTER
011620  005337  001200          1$:     DEC     @#$TMP1         ;WAIT FOR INTERRUPT
011624  001375                  BNE     1$                      ;BRANCH IF NOT ZERO
                                                                ;BEFORE THIS IS ZERO INTERRUPT WHOULD
                                                                ;OCCUR
011626  012712  000040                  MOV     #40,@R2         ;CLEAR THE CONTROLLER VIA CS2 CLR BIT
011632  000404                  BR      TST10   ;       NO INTERRUPT SO CONTINUE ----------------)

011634  022626                  RPTRP2: CMP     (SP)+,(SP)+     ;RESTORE STACK
011636  104065                  ERROR   65                      ;INTERRUPT OCCURRED WITH PROCESSOR
                                                                ;PROCESSOR STATUS SAME AS DISK
011640  012712  000040                  MOV     #40,@R2         ;CLEAR THE CONTROLLER VIA CS2 CLR BIT
```

```
                                ;;*****************************************************************
                                ;*TEST 10       PACK ACKNOWLEDGE
                                ;*
                                ;*      IF STARTING ADDRESS 220 IS USED THIS TEST WILL NOT BE PERFORMED
                                ;*
                                ;*      IF THE PROGRAM WORKS UNDER ACT-11 MONITOR
                                ;*      THEN THIS TEST IS NOT PERFORMED
                                ;*
                                ;*      IF NO ACT-11 MONITOR IS PRESENT
                                ;*      THEN THIS TEST IS PERFORMED ONLY ON THE FIRST PASS
                                ;*      ON SUBSEQUENT PASSES THIS TEST IS NOT DONE
                                ;*
                                ;*      THIS TESTS THE ACKNOWLEDGE COMMAND=44
                                ;*      VV BIT - RHDS1 BIT #6
                                ;*      MOL BIT - RHDS1 BIT #12
                                ;*      DVA BIT - RHCS1 BIT #11
                                ;*      THE DRIVE IS STOPED MOL IS CHECKED TO BE 0
                                ;*      AND DVA SHOULD BE 0
                                ;*      THE DRIVE IS POWERED UP
                                ;*      VV SHOULD BE 0, MOL SHOULD BE 1, DVA SHOULD BE 1
                                ;*      GO SHOULD BE 0
                                ;*      ALL REGISTERS EXCEPT RHDB, RHLA AND RHCC ARE STORED
                                ;*      PACK ACKNOWLEDGE IS ISSUED
                                ;*      ALL STORED REGISTERS SHOULD BE UNCHANGED
                                ;*      EXCEPT VV
                                ;*
                                ;;*****************************************************************
011644  000004                  TST10:  SCOPE
011646  012706  001000                  MOV     #STACK,SP       ;RESET STACK
011652  012737  000010  004604          MOV     #10,@#TSTNM     ;SAVE TEST NUMBER
                                ;
011660  004737  041524                  JSR     PC,@#CLDISK     ;SET R1-RHCS1, R2-RHCS2
                                                                ;R3-RHDS1, R4-RHER1
                                                                ;GIVE RH-11 INITIALIZE
                                                                ;SETUP UNIT NUMBER
011664  012737  000000  177776          MOV     #0,PS           ;SET PROCESSER STATUS TO 0
                                ;
                                ;*THIS CODE CHECKS TO SEE IF MANUAL INTERVENTION TESTS ARE OK
                                ;
011672  005737  004724                  TST     @#NOPUSH        ;IS THIS A 220 START ?
011676  001007                          BNE     1$              ;SKIP THIS TEST IF SO
011700  005737  000042                  TST     @#42            ;MONITOR (ACT 11) RETURN ADDRESS ?
011704  001004                          BNE     1$              ;SKIP THIS TEST
011706  005737  001100                  TST     @#$PASS         ;FIRST PASS ?
011712  001001                          BNE     1$              ;SKIP THIS TEST IF NOT
011714  000402                          BR      2$              ;CONTINUE WITH THIS TEST
                                ;
011716                          1$:
011716  000137  012424                  JMP     TST11   ;       JUMP TO NEXT TEST ----------------)
011722                          2$:
011722  104401  011730                  TYPE    ,65$            ;;TYPE ASCIZ STRING
011726  000407                          BR      64$             ;;GET OVER THE ASCIZ
                                ;;65$:  .ASCIZ  <15><12>/STOP DRIVE /
011746                          64$:
011746  013746  004716                  MOV     @#UNIT,-(SP)    ;GET UNIT UNDER TEST
```

```
                                        TYPDS
                                        TYPE    ,67$            ;;TYPE ASCIZ STRING
                                        BR      66$             ;;GET OVER THE ASCIZ
                                ;;67$:  .ASCIZ  / THEN HIT CONTINUE/<15><12>
                                66$:
                                        HALT
                                        BIT     #MOL,@R3        ;MOL IN RHDS1 SHOULD BE = 0
                                        BEQ     3$              ;BRANCH IF MOL=0
                                        MOV     R3,@#$BDADR     ;FAILING REGISTER ADDRESS-RHDS1
                                        ERROR   10              ;ON SPINPLE POWERED DOWN
                                                                ;MOL SHOULD BE 0
                                3$:     MOV     @#UNIT,-(SP)    ;UNIT NUMBER
                                        BIS     #IR,(SP)        ;INCLUDE IR
                                        CMP     (SP)+,@R2       ;ONLY UNIT NO. AND IR SHOULD BE
                                                                ;HIGH IN RHCS2
                                        BEQ     4$              ;BRANCH IF RHCS2 GOOD
                                        MOV     R2,@#$BDADR     ;FAILING REGISTER ADDRESS-RHCS2
                                        ERROR   11              ;WITH SPINDLE POWERED DOWN
                                                                ;ONLY UNIT NO. AND IR SHOULD BE
                                                                ;HIGH

                                4$:

                                        JSR     PC,@#CLDISK     ;SET R1-RHCS1, R2-RHCS2
                                                                ;R3-RHDS1, R4-RHER1
                                                                ;GIVE RH-11 INITIALIZE
                                                                ;SETUP UNIT NUMBER
                                        TYPE    ,69$            ;;TYPE ASCIZ STRING
                                        BR      68$             ;;GET OVER THE ASCIZ
                                ;;69$:  .ASCIZ  <15><12>/START DRIVE/
                                68$:
                                        MOV     @#UNIT,-(SP)    ;GET UNIT UNDER TEST
                                        TYPDS
                                        TYPE    ,71$            ;;TYPE ASCIZ STRING
                                        BR      70$             ;;GET OVER THE ASCIZ
                                ;;71$:  .ASCIZ  / AFTER HEAD LOAD HIT CONTINUE/<15><12>
                                70$:
                                        HALT
                                        BIT     #MOL,@R3        ;MOL IN RHDS1 SHOULD BE = 1
                                        BEQ     5$              ;BRANCH IF MOL = 0
                                        BIT     #DPR,@R3        ;DPR IN RHDS1 SHOULD BE = 1
                                        BEQ     5$              ;BRANCH IF DPR = 0
                                        BIT     #DRY,@R3        ;DRY IN RHDS1 SHOULD BE = 1
                                        BEQ     5$              ;BRANCH IF DRY =0
                                        BIT     #VV,@R3         ;VV IN RHDS1 SHOULD BE = 0
                                        BEQ     6$              ;BRANCH IF VV = 0 (GOOD)
                                5$:     MOV     R3,@#$BDADR     ;FAILING REGISTER ADDRESS - RHDS1
                                        ERROR   12              ;WITH SPINDLE POWERED UP
                                                                ;RHDS1 SHOULD HAVE VV = 0, MOL = 1

                                6$:     MOV     @R1,R0          ;GET RHCS1 CONTENTS
                                        BIC     #SC!TRE!MCPE!76,R0 ;CLEAR SC,TRE,MCPE AND
                                                                ;ALL FUNCTION BITS
                                        CMP     #DVA!RDY,R0     ;RHCS1 SHOULD HAVE
                                                                ;GO = 0, DVA = 1, RDY = 1
                                        BEQ     7$              ;BRANCH IF RHCS1 IS GOOD
```

```
 2914  012230  010137  001122                MOV     R1,@#$BDADR     ;FAILING REGISTER RHCS1
 2915  012234  104013                        ERROR   13              ;AFTER A POWER UP WITHOUT ANY
 2916                                                                 ;INIT RHCS1 SHOULD HAVE
 2917                                                                 ;GO = 0, DVA = 1, RDY = 1, IE = 0
 2918                                                                 ;DISREGARD ALL OTHER BITS
 2919
 2920  012236  005777  170072          7$:   TST     @RHCC           ;TEST RHCC, IT SHOULD = 0
 2921  012242  001411                        BEQ     10$             ;BRANCH IF RHCC = 0
 2922  012244  013737  002334  004600        MOV     @#RHCC,@#REGADR ;FAILING REGISTER RHCC
 2923  012252  005037  001124                CLR     @#$GDDAT        ;RHCC SHOULD BE = 0
 2924  012256  017737  170052  001126        MOV     @RHCC,@#$BDDAT  ;BAD DATA
 2925  012264  104014                        ERROR   14              ;AFTER POWER UP RHCC
 2926                                                                 ;SHOULD BE 0
 2927
 2928  012266                          10$:
 2929
 2930  012266  004737  041524                JSR     PC,@#CLDISK     ;SET R1-RHCS1, R2-RHCS2
 2931                                                                 ;R3-RHDS1, R4-RHER1
 2932                                                                 ;GIVE RH-11 INITIALIZE
 2933                                                                 ;SETUP UNIT NUMBER
 2934  012272  004737  041562                JSR     PC,@#CHECK      ;CHECK THAT DVA,RDY,MOL,DPR,DRY = 1
 2935                                                                 ;AND THAT NO STATUS BITS ARE STUCK = 1
 2936  012276  104401  067033                TYPE    ,CPHALT         ;CANNOT CONTINUE TESTS IF THEY AREN'T
 2937  012302  000000                        HALT                    ;STOP
 2938  012304  013777  002460  167766        MOV     @#PKACK,@RHCS1  ;GET READY FOR PKACK
 2939                                                                 ;PACK ACKNOWLEGDE WITH 22 IN RHCS1
 2940
 2941
 2942                                  ;*NOW SAVE REGISTERS FOR COMPARISON AFTER PACK ACKNOWLEDGE
 2943
 2944  012312  004037  041672                JSR     R0,@#SAVER      ;SAVE REGISTERS
 2945  012316  002272                        RHWC                    ;RHWC IS THE FIRST REGISTER SAVED
 2946  012320  004612                        SAVERE                  ;STARTING ADDRESS OF WHERE
 2947                                                                 ;THE REGISTERS ARE SAVED
 2948  012322  000022                        18.                     ;NUMBER OF REGISTERS
 2949                                                                 ;SAVED = 18.
 2950
 2951  012324  013777  004606  167734        MOV     @#RP4VEC,@RPVEC ;SET RP04 VECTOR ADDRESS
 2952                                                                 ;TO 'TIME1' IF P-CLOCK IS PRESENT
 2953                                                                 ;OR TO 'TIME2' IF P-CLOCK IS NOT PRESENT
 2954                                                                 ;'TIME' WILL ONLY SAVE
 2955                                                                 ;CURRENT CYLINDER ADDRESS
 2956                                                                 ;AND LOOK AHEAD REGISTERS
 2957
 2958
 2959  012332  013746  002460                MOV     @#PKACK,-(SP)   ;GET READY TO MOVE COMMAND
 2960  012336  052716  000001                BIS     #GO,(SP)        ;GET READY TO SET GO
 2961                                                                 ;WITHOUT INTERRUPT ENABLE
 2962  012342  012677  167732                MOV     (SP)+,@RHCS1    ;GO WITH
 2963                                                                  ;22 IN RHCS1 FOR PACK ACKNOWLEDGE
 2964                                                                 ;WITH INTERRUPT DISABLED
 2965
 2966
 2967
 2968  012346  104413                        WAT                     ;WAIT FOR VV BIT TO SET
 2969  012350  002322                        RHDS1                   ;WAIT FOR RHDS1 REGISTER
```

```
2970  012352  000100                          VV                      ;WAIT FOR VV BIT IN RHDS1 REGISTER
2971  012354  000001                          1.                      ;ALLOW 10 MICRO SECONDS
2972  012356  000001                          1.                      ;VV MUST SET BETWEEN
2973                                                                  ;00 AND 20 MICRO SECONDS
2974
2975
2976  012360  004037  042404                  JSR     R0,@#CHREG      ;CHANGE BITS IN SAVED REGISTER
2977  012364  002322                          RHDS1                   ;CHANGE RHDS1 REGISTER
2978
2979  012366  000001                          1                       ;1 BIT/BITS TO BE CHANGED
2980  012370  000001                          1                       ;NEW VALUE OF VV IS 1
2981  012372  000100                          VV                      ;CHANGE VV BIT
2982
2983                                          ;*NOW COMPARE REGISTERS SO THAT NO REGISTERS
2984                                          ;*CHANGED EXCEPT VV IN RHDS1 AND IE IN RHCS1
2985
2986
2987  012374  004037  042512                  JSR     R0,@#COMREG     ;COMPARE SAVED REGISTERS WITH
2988                                                                  ;PRESENT VALUE
2989  012400  004612                          SAVERE                  ;GOOD DATA SAVED IN 'SAVERE'
2990  012402  002354                          WC                      ;TEST DATA STARTING FROM 'RHWC'
2991  012404  000022                          18.                     ;18. REGISTERS TO BE COMPARED
2992  012406  012412                          11$                     ;RETURN TO 11$ ON ERROR
2993  012410  012416                          12$                     ;RETURN TO 12$ ON NO ERROR
2994
2995  012412  104015                  11$:    ERROR   15              ;GIVING A PACK ACKNOWLEDGE
2996  012414  000207                          RTS     PC              ;CAUSED AN ERROR
2997                                                                  ;PACK ACKNOWLEDGE SHOULD
2998                                                                  ;SET VV IN RHDS1
2999                                                                  ;INTERRUPT SHOULD MAKE
3000                                                                  ;IE = 0
3001                                                                  ;NO OTHER REGISTERS SHOULD
3002                                                                  ;CHANGE
3003                                                                  ;GOOD DATA GIVES
3004                                                                  ;CONTENTS OF REGISTER BEFORE
3005                                                                  ;PACK ACKNOWLEDGE
3006                                                                  ;RECEIVED DATA GIVES
3007                                                                  ;CONTENTS OF REGISTER
3008                                                                  ;AFTER PACK ACKNOWLEDGE
3009
3010  012416  012737  177777  047342  12$:    MOV     #-1,@#PRITEM    ;CLEAR PREVIOUS ITEM NUMBER
3011
```

```
3012
3013
3014
3015                                    ;;*****************************************************************
3016                                    ;*TEST 11       SET VV BIT #6 IN RHDS1
3017                                    ;*      THIS TEST SETS VV IN RHDS1 INCASE
3018                                    ;*      ACT-11 MONITOR IS PRESENT AND THE PREVIOUS TEST
3019                                    ;*      IS NOT PERFORMED
3020                                    ;*      THERE IS A RESET AT THE BEGINING OF THIS TEST
3021                                    ;*      FOR ERROR RECOVERY ONLY.
3022
3023                                    ;;*****************************************************************
3024  012424  000004                    TST11:  SCOPE
3025
3026                                            ;*IN CASE THERE IS ANY DRIVE ERRORS DURING POWER UP
3027                                            ;*OR POWER DOWN OR ANY PARITY ERRORS A RESET IS GIVEN
3028  012426  000005                    RESET
3029  012430  004737  045734            JSR     PC,@#$TKINT     ;INITILIZE TK
3030
3031
3032  012434  012706  001000            MOV     #STACK,SP       ;RESET STACK
3033  012440  012737  000011  004604    MOV     #11,@#TSTNM     ;SAVE TEST NUMBER
3034
3035  012446  004737  041524            JSR     PC,@#CLDISK     ;SET R1-RHCS1, R2-RHCS2
3036                                                            ;R3-RHDS1, R4-RHER1
3037                                                            ;GIVE RH-11 INITIALIZE
3038                                                            ;SETUP UNIT NUMBER
3039  012452  004737  041562            JSR     PC,@#CHECK      ;CHECK THAT DVA,RDY,MOL,DPR,DRY = 1
3040                                                            ;AND THAT NO STATUS BITS ARE STUCK = 1
3041  012456  104401  067033            TYPE    ,CPHALT         ;CANNOT CONTINUE TESTS IF THEY AREN'T
3042  012462  000000                    HALT                    ;STOP
3043  012464  013777  002460  167606    MOV     @#PKACK,@RHCS1  ;GET READY FOR PKACK
3044                                                            ;PACK ACKNOWLEGDE WITH 22 IN RHCS1
3045
3046
3047                                    ;*NOW SAVE REGISTERS FOR COMPARISON AFTER PACK ACKNOWLEDGE
3048
3049  012472  004037  041672            JSR     R0,@#SAVER      ;SAVE REGISTERS
3050  012476  002272                    RHWC                    ;RHWC IS THE FIRST REGISTER SAVED
3051  012500  004612                    SAVERE                  ;STARTING ADDRESS OF WHERE
3052                                                            ;THE REGISTERS ARE SAVED
3053  012502  000022                    18.                     ;NUMBER OF REGISTERS
3054                                                            ;SAVED = 18.
3055
3056  012504  013777  004606  167554    MOV     @#RP4VEC,@RPVEC ;SET RP04 VECTOR ADDRESS
3057                                                            ;TO 'TIME1' IF P-CLOCK IS PRESENT
3058                                                            ;OR TO 'TIME2' IF P-CLOCK IS NOT PRESENT
3059                                                            ;'TIME' WILL ONLY SAVE
3060                                                            ;CURRENT CYLINDER ADDRESS
3061                                                            ;AND LOOK AHEAD REGISTERS
3062
3063
3064  012512  013746  002460            MOV     @#PKACK,-(SP)   ;GET READY TO MOVE COMMAND
3065  012516  052716  000001            BIS     #GO,(SP)        ;GET READY TO SET GO
3066                                                            ;WITHOUT INTERRUPT ENABLE
3067  012522  012677  167552            MOV     (SP)+,@RHCS1    ;GO WITH
```

```
3068                                                    ;22 IN RHCS1 FOR PACK ACKNOWLEDGE
3069                                                    ;WITH INTERRUPT DISABLED
3070
3071  012526  011100                    MOV     @R1,R0          ;SAVE RHCS1 DURING ABOVE OPERATION
3072  012530  011305                    MOV     @R3,R5          ;SAVE RHDS1 DURING ABOVE OPERATION
3073
3074
3075  012532  104413                    WAT                     ;WAIT FOR VV BIT TO SET
3076  012534  002322                    RHDS1                   ;WAIT FOR RHDS1 REGISTER
3077  012536  000100                    VV                      ;WAIT FOR VV BIT IN RHDS1 REGISTER
3078  012540  000001                    1.                      ;ALLOW 10 MICRO SECONDS
3079  012542  000001                    1.                      ;VV MUST SET BETWEEN
3080                                                            ;00 AND 20 MICRO SECONDS
3081
3082                                    ;*COMPARE CONTENTS OF RHCS1 AND RHDS1 ALREADY SAVED IN
3083                                    ;*R0 AND R5 IMMEDIATELY AFTER GO
3084
3085  012544  013746  002460           MOV     @#PKACK,-(SP)   ;SAVE COMMAND
3086  012550  052716  004200           BIS     #DVA!RDY,(SP)   ;INCLUDE DVA!RDY
3087  012554  011637  001124           MOV     (SP),@#$GDDAT   ;SAVE FOR PRINTOUT
3088  012560  022600                    CMP     (SP)+,R0        ;DURING ABOVE OPERATION ONLY DVA!RDY
3089                                                            ;AND COMMAND SHOULD BE SET
3090  012562  001405                    BEQ     64$             ;BRANCH IF GOOD
3091  012564  010037  001126           MOV     R0,@#$BDDAT     ;BAD DATA
3092  012570  010137  004600           MOV     R1,@#REGADR     ;FAILING REGISTER RHCS1
3093  012574  104021                    ERROR   21              ;DURING ABOVE OPERATION ONLY
3094                                                            ;COMMAND AND DVA!RDY SHOULD BE SET
3095  012576  012746  010700    64$:   MOV     #MOL!DPR!DRY!VV,-(SP)   ;SAVE BITS SET DURING OPERATION IN RHDS1
3096  012602  011637  001124           MOV     (SP),@#$GDDAT   ;SAVE FOR PRINTOUT
3097  012606  022605                    CMP     (SP)+,R5        ;DURING ABOVE OPERATION ONLY MOL!DPR!DRY!VV
3098                                                            ;SHOULD BE SET
3099  012610  001405                    BEQ     66$             ;BRANCH IF GOOD
3100  012612  010537  001126           MOV     R5,@#$BDDAT     ;BAD DATA
3101  012616  010337  004600           MOV     R3,@#REGADR     ;FAILING REGISTER RHDS1
3102  012622  104063                    ERROR   63              ;DURING ABOVE OPERATION ONLY
3103                                                            ;MOL!DPR!DRY!VV SHOULD BE SET
3104  012624                    66$:
3105
3106  012624  004037  042404           JSR     R0,@#CHREG      ;CHANGE BITS IN SAVED REGISTER
3107  012630  002322                    RHDS1                   ;CHANGE RHDS1 REGISTER
3108
3109  012632  000001                    1                       ;1 BIT/BITS TO BE CHANGED
3110  012634  000001                    1                       ;NEW VALUE OF VV IS 1
3111  012636  000100                    VV                      ;CHANGE VV BIT
3112
3113                                    ;*NOW COMPARE REGISTERS BEFORE PACK ACKNOWLEDGE
3114                                    ;*WITH AFTER PACK ACKNOWLEDGE
3115
3116
3117
3118  012640  004037  042512           JSR     R0,@#COMREG     ;COMPARE SAVED REGISTERS WITH
3119                                                            ;PRESENT VALUE
3120  012644  004612                    SAVERE                  ;GOOD DATA SAVED IN 'SAVERE'
3121  012646  002354                    WC                      ;TEST DATA STARTING FROM 'RHWC'
3122  012650  000022                    18.                     ;18. REGISTERS TO BE COMPARED
3123  012652  012656                    1$                      ;RETURN TO 1$ ON ERROR
```

```
3124  012654  012662                    2$                  ;RETURN TO 2$ ON NO ERROR
3125
3126  012656  104015              1$:   ERROR   15          ;GIVING A PACK ACKNOWLEDGE
3127  012660  000207                    RTS     PC          ;CAUSED AN ERROR
3128                                                        ;PACK ACKNOWLEDGE SHOULD
3129                                                        ;SET VV IN RHDS1
3130                                                        ;INTERRUPT SHOULD MAKE
3131                                                        ;IE = 0
3132                                                        ;NO OTHER REGISTERS SHOULD
3133                                                        ;CHANGE
3134                                                        ;GOOD DATA GIVES CONTENTS
3135                                                        ;OF REGISTER BEFORE COMMAND
3136                                                        ;RECEIVED DATA GIVES CONTENTS
3137                                                        ;OF REGISTER AFTER COMMAND
3138  012662                        2$:
```

```
3139
3140                                    ;;*****************************************************************
3141  012662  005737  004744                      TST     @#RP06          ;TEST FOR RP06 DRIVE
3142  012666  001005                              BNE     3$              ;IF = 1 DO "MAKECYL" 777
3143  012670  005737  004746                      TST     @#RP05          ;TEST FOR RP05 DRIVE
3144  012674  001004                              BNE     4$              ;IF = 1 DO "MAKECYL" 377
3145  012676  000137  015022                      JMP     TST16   ;       IF BOTH = 0, DON'T DO EITHER "MAKECYL" ------)
3146                                                                      ;OR THE ADDRESS PLUG TESTS
3147  012702                                  3$:
3148  012702  000137  012746                      JMP     TST13   ;       DO "MAKECYL" 777 --------------------------)
3149  012706                                  4$:
3150                                    ;;*****************************************************************
3151
3152
3153
3154                                    ;;*****************************************************************
3155                                    ;*TEST 12        MAKE CURRENT CYLINDER = 377
3156                                    ;;*****************************************************************
3157  012706  000004                    TST12:  SCOPE
3158  012710  012706  001000                      MOV     #STACK,SP       ;RESET STACK
3159  012714  012737  000012  004604              MOV     #13-1,@#TSTNM   ;THIS SAVES TEST NUMBER
3160
3161  012722  004737  041524                      JSR     PC,@#CLDISK     ;INIT DRIVE
3162  012726  012777  000001  167364              MOV     #DMD,@RHMR      ;SET DIAGNOSTIC MODE
3163  012734  004037  041174                      JSR     R0,@#MAKECYL    ;SUBROUTINE TO GIVE A SEEK
3164                                                                      ;COMMAND FOLLOWED BY A INIT
3165                                                                      ;THIS SHOULD CHANGE RHCC TO 377
3166  012740  000377                              377
3167
3168
3169  012742  000137  013002                      JMP     TST14   ;       SKIP NEXT "MAKECYL" ----------------------)
3170
3171
3172
3173                                    ;;*****************************************************************
3174                                    ;*TEST 13        MAKE CURRENT CYLINDER = 777
3175                                    ;;*****************************************************************
3176  012746  000004                    TST13:  SCOPE
3177  012750  012706  001000                      MOV     #STACK,SP       ;RESET STACK
3178  012754  012737  000013  004604              MOV     #14-1,@#TSTNM   ;THIS SAVES TEST NUMBER
3179
3180  012762  004737  041524                      JSR     PC,@#CLDISK     ;INIT DRIVE
3181  012766  012777  000001  167324              MOV     #DMD,@RHMR      ;SET DIAGNOSTIC MODE
3182  012774  004037  041174                      JSR     R0,@#MAKECYL    ;SUBROUTINE TO GIVE A SEEK
3183                                                                      ;COMMAND FOLLOWED BY A INIT
3184                                                                      ;THIS SHOULD CHANGE RHCC TO 777
3185  013000  000777                              777
3186
3187
```

```
3188
3189
3190                                    ;;*****************************************************************
3191                                    ;*TEST 14       ADDRESS PLUG CHANGE ERROR
3192
3193                                    ;*      CHECK PROPER ADDRESS PLUG FUNCTIONALITY
3194                                    ;*      BY PULLING THE ADDRESS PLUG DURING A COMMAND ISSUED
3195                                    ;*      IN DIAGNOSTIC MODE (TO GUARANTEE COMMAND IS STILL ACTIVE WHEN
3196                                    ;*      PLUG IS PULLED) AND VERIFYING THAT THE DRIVE GOES OFF LINE
3197
3198                                    ;*      THE ADDRESS PLUG IS THEN REPLACED AND RETURN OF THE PROPER
3199                                    ;*      RESPONDING DRIVE IS VERIFIED, AS WELL AS THE FACT THAT
3200                                    ;*      ATTENTION BITS COME UP IN THE PROPER BIT LOCATION(?????), AND THE
3201                                    ;*      ADDRESS CHANGE ERROR (ACE) - RHER3 BIT #13 IS SET, AS WELL AS
3202                                    ;*      SC,TRE,ERR,AND MCPE; VV IS RESET AND RHCC = 000 (DRIVE RECALIBRATED)
3203
3204                                    ;*      IN ADDITION VERIFICATION IS ALSO MADE THAT NO OTHER DRIVE
3205                                    ;*      NUMBERS APPEAR ON THE BUSS AFTER THE PLUG IS REPLACED. (??????)
3206
3207                                    ;;*****************************************************************
3208  013002  000004                   TST14:  SCOPE
3209  013004  012737  000001  001212           MOV     #1,$TIMES       ;;DO 1 ITERATION
3210
3211                                    ;;*****************************************************************
3212  013012  005737  004744                   TST     @#RP06          ;TEST FOR RP06 DRIVE
3213  013016  001005                           BNE     4$              ;IF = 1, DO THIS TEST
3214  013020  005737  004746                   TST     @#RP05          ;TEST FOR MEMOREX RP04
3215  013024  001002                           BNE     4$              ;IF = 1, DO THIS TEST
3216                                                                    ;IF NEITHER FLAG IS UP, ASSUME THE
3217                                                                    ;DRIVE IS AN ISS RP04 AND SKIP TEST
3218  013026  000137  014014                   JMP     TST15   ;       JUMP TO NEXT TEST ----------------)
3219  013032                           4$:
3220                                    ;;*****************************************************************
3221
3222  013032  012706  001000                   MOV     #STACK,SP       ;RESET STACK
3223  013036  012737  000014  004604           MOV     #14,@#TSTNM     ;SAVE TEST NUMBER
3224
3225  013044  004737  041524                   JSR     PC,@#CLDISK     ;SET R1-RHCS1, R2-RHCS2
3226                                                                    ;R3-RHDS1, R4-RHER1
3227                                                                    ;GIVE RH-11 INITIALIZE
3228                                                                    ;SETUP UNIT NUMBER
3229  013050  004737  041604                   JSR     PC,@#CHECKT     ;CHECK DVA,RDY,MOL,DPR,DRY,VV = 1
3230                                                                    ;AND THAT NO STATUS BITS ARE STUCK = 1
3231  013054  104401  067033                   TYPE    ,CPHALT         ;CANNOT CONTINUE TESTING IF ANY OF
3232                                                                    ;THE FIRST SET OF BITS DON'T = 1
3233  013060  000000                           HALT                    ;STOP
3234
3235                                    ;*THIS CODE CHECKS TO SEE IF MANUAL INTERVENTION TESTS ARE OK
3236
3237  013062  005737  004724                   TST     @#NOPUSH        ;IS THIS A 220 START ?
3238  013066  001007                           BNE     1$              ;SKIP THIS TEST IF SO
3239  013070  005737  000042                   TST     @#42            ;MONITOR (ACT 11) RETURN ADDRESS ?
3240  013074  001004                           BNE     1$              ;SKIP THIS TEST
3241  013076  005737  001100                   TST     @#$PASS         ;FIRST PASS ?
3242  013102  001001                           BNE     1$              ;SKIP THIS TEST IF NOT
3243  013104  000402                           BR      2$              ;CONTINUE WITH THIS TEST
```

```
3244
3245  013106                          1$:
3246  013106  000137  014014                  JMP     TST15   ;       JUMP TO NEXT TEST ----------------)
3247  013112                          2$:
3248
3249                                          ;*SET DIAGNOSTIC MODE TO ENABLE A COMMAND ACTIVE WHILE
3250                                          ;*THE PLUG IS PULLED
3251
3252  013112  052777  000001  167200          BIS     #DMD,@RHMR      ;SET UP DIAGNOSTIC MODE
3253
3254                                          ;*TAKE AN INITIAL REGISTER SNAPSHOT
3255
3256  013120  004037  041672                  JSR     R0,@#SAVER      ;SAVE REGISTERS
3257  013124  002272                          RHWC                    ;RHWC IS THE FIRST REGISTER SAVED
3258  013126  004612                          SAVERE                  ;STARTING ADDRESS OF WHERE
3259                                                                  ;THE REGISTERS ARE SAVED
3260  013130  000022                          18.                     ;NUMBER OF REGISTERS
3261                                                                  ;SAVED = 18.
3262
3263
3264                                          ;*ISSUE A COMMAND AND THE 'GO' BIT (NOT POSITIONING COMMAND)
3265                                          ;*TO VERIFY COMMAND ABORT IF PLUG IS PULLED
3266
3267
3268
3269                                          ;*ISSUE SOME CLOCKS TO GET THE COMMAND STARTED
3270                                          ;*(USE "SEARCH" WITH "DTETST" FLAG UP TO STOP CLOCKING ?)
3271
3272
3273
3274  013132  104401  013140                  TYPE    ,65$            ;;TYPE ASCIZ STRING
3275  013136  000420                          BR      64$             ;;GET OVER THE ASCIZ
3276                                  ;;65$:  .ASCIZ  <15><12>/REMOVE  ADDRESS PLUG ON DRIVE/
3277  013200                          64$:
3278  013200  013746  004716                  MOV     @#UNIT,-(SP)    ;GET THE UNIT NO. UNDER TEST
3279  013204  104405                          TYPDS                   ;TYPE IT OUT
3280  013206  104401  013214                  TYPE    ,67$            ;;TYPE ASCIZ STRING
3281  013212  000415                          BR      66$             ;;GET OVER THE ASCIZ
3282                                  ;;67$:  .ASCIZ  /       THEN PRESS CONTINUE/
3283  013246                          66$:
3284  013246  000000                          HALT                    ;WAIT FOR OPERATOR PLUG CHANGE
3285
3286                                          ;*CHECK THAT THE UNIT NO. UNDER TEST HAS GONE OFFLINE
3287
3288  013250  017700  167030                  MOV     @RHDST,R0       ;ATTEMPT TO ADDRESS THE DRIVE
3289  013254  004737  043436                  JSR     PC,@#PUTREG     ;TAKE REGISTER SNAPSHOTS
3290  013260  032737  010000  002360          BIT     #NED,@#CS2      ;TEST FOR NON EXISTENT DRIVE
3291  013266  001001                          BNE     7$              ;CONTINUE IF 'NED' BIT SET (UNIT
3292                                                                  ;IS OFFLINE AS IT SHOULD BE)
3293  013270  104101                          ERROR   101             ;UNIT DID NOT GO OFFLINE WHEN ADDRESS
3294                                                                  ;PLUG WAS REMOVED
3295  013272                          7$:
```

```
3296
3297                                    ;*NOW REPLACE THE ADDRESS PLUG
3298
3299
3300  013272  104401  013300            TYPE    ,69$            ;;TYPE ASCIZ STRING
3301  013276  000420                    BR      68$             ;;GET OVER THE ASCIZ
3302                            ;;69$:  .ASCIZ  <15><12>/REPLACE ADDRESS PLUG ON DRIVE/
3303  013340                    68$:
3304  013340  013746  004716            MOV     @#UNIT,-(SP)    ;GET THE UNIT UNDER TEST
3305  013344  104405                    TYPDS                   ;TYPE IT OUT
3306  013346  104401  013354            TYPE    ,71$            ;;TYPE ASCIZ STRING
3307  013352  000416                    BR      70$             ;;GET OVER THE ASCIZ
3308                            ;;71$:  .ASCIZ  /      THEN PRESS CONTINUE/<15><12>
3309  013410                    70$:
3310  013410  000000                    HALT                    ;WAIT FOR OPERATOR PLUG REPLACEMENT
3311
3312                                    ;*CHECK THAT THE ORIGINAL UNIT HAS COME BACK ON LINE
3313
3314  013412  004737  043436            JSR     PC,@#PUTREG     ;TAKE NEW REGISTER SNAPSHOTS
3315  013416  032737  000400  002404    BIT     #DPR,@#DS1      ;TEST THAT 'DPR' = 1
3316  013424  001411                    BEQ     9$              ;ERROR - DRIVE SHOULD BE PRESENT
3317  013426  032737  004000  002362    BIT     #DVA,@#CS1      ;TEST THAT DEVICE IS NOW AVAILABLE
3318  013434  001405                    BEQ     9$              ;ERROR - 'DVA' SHOULD = 1
3319  013436  032737  000200  002404    BIT     #DRY,@#DS1      ;TEST THAT DRIVE READY IS = 1
3320  013444  001401                    BEQ     9$              ;ERROR - 'DRY' SHOULD = 1
3321  013446  000403                    BR      8$              ;A-OK: 'DPR' = 1, 'DVA' = 1, & 'DRY' =1
3322  013450  104102            9$:     ERROR   102             ;UNIT NOT AVAILABLE AFTER
3323                                                            ;ADDRESS PLUG WAS REPLACED
3324  013452  000137  014014            JMP     TST15   ;       JUMP TO NEXT TEST ---------------)
3325
3326  013456                    8$:     ;CHANGE THE INITIAL REGISTER SNAPSHOT TO EXPECTED VALUES
3327                                    ;*AFTER THE PLUG CHANGE
3328
3329
3330  013456  004037  042404            JSR     R0,@#CHREG      ;CHANGE BITS IN SAVED REGISTER
3331  013462  002300                    RHCS1                   ;CHANGE RHCS1 REGISTER
3332
3333  013464  000004                    4                       ;4 BIT/BITS TO BE CHANGED
3334  013466  000001                    1                       ;NEW VALUE OF RDY IS 1
3335  013470  000200                    RDY                     ;CHANGE RDY BIT
3336  013472  000001                    1                       ;NEW VALUE OF SC IS 1
3337  013474  100000                    SC                      ;CHANGE SC BIT
3338  013476  000001                    1                       ;NEW VALUE OF TRE IS 1
3339  013500  040000                    TRE                     ;CHANGE TRE BIT
3340  013502  000001                    1                       ;NEW VALUE OF MCPE IS 1
3341  013504  020000                    MCPE                    ;CHANGE MCPE BIT
3342
3343  013506  004037  042404            JSR     R0,@#CHREG      ;CHANGE BITS IN SAVED REGISTER
3344  013512  002322                    RHDS1                   ;CHANGE RHDS1 REGISTER
3345
3346  013514  000003                    3                       ;3 BIT/BITS TO BE CHANGED
3347  013516  000001                    1                       ;NEW VALUE OF ATA IS 1
3348  013520  100000                    ATA                     ;CHANGE ATA BIT
3349  013522  000001                    1                       ;NEW VALUE OF ERR IS 1
3350  013524  040000                    ERR                     ;CHANGE ERR BIT
3351  013526  000000                    0                       ;NEW VALUE OF VV IS 0
```

```
3352  013530  000100                        VV                          ;CHANGE VV BIT
3353
3354  013532  004037  042404                JSR     R0,@#CHREG          ;CHANGE BITS IN SAVED REGISTER
3355  013536  002276                        RHCS2                       ;CHANGE RHCS2 REGISTER
3356
3357  013540  000001                        1                           ;1 BIT/BITS TO BE CHANGED
3358  013542  000001                        1                           ;NEW VALUE OF NED IS 1
3359  013544  010000                        NED                         ;CHANGE NED BIT
3360
3361  013546  004037  042404                JSR     R0,@#CHREG          ;CHANGE BITS IN SAVED REGISTER
3362  013552  002314                        RHER3                       ;CHANGE RHER3 REGISTER
3363
3364  013554  000001                        1                           ;1 BIT/BITS TO BE CHANGED
3365  013556  000001                        1                           ;NEW VALUE OF ACE IS 1
3366  013560  020000                        ACE                         ;CHANGE ACE BIT
3367
3368  013562  053737  004740  004636        BIS     @#ATTENT,@#SAVERE+24 ;SET UNIT UNDER TEST ATTENTION
3369  013570  005037  004654                CLR     @#SAVERE+42         ;SET RHCC REGISTER IMAGE TO ALL 0'S
3370
3371                                        ;*TAKE A NEW REGISTER SNAPSHOT AND
3372                                        ;*COMPARE THE REGISTER CONTENTS WITH EXPECTED VALUES
3373
3374
3375  013574  004037  042512                JSR     R0,@#COMREG         ;COMPARE SAVED REGISTERS WITH
3376                                                                      ;PRESENT VALUE
3377  013600  004612                        SAVERE                      ;GOOD DATA SAVED IN 'SAVERE'
3378  013602  002354                        WC                          ;TEST DATA STARTING FROM 'RHWC'
3379  013604  000022                        18.                         ;18. REGISTERS TO BE COMPARED
3380  013606  013612                        5$                          ;RETURN TO 5$ ON ERROR
3381  013610  013616                        6$                          ;RETURN TO 6$ ON NO ERROR
3382
3383  013612  104100              5$:       ERROR   100                 ;ADDRESS PLUG CHANGE CAUSED SOME
3384                                                                      ;REGISTER ERROR
3385  013614  000207                        RTS     PC                  ;GO BACK AND CHECK THE NEXT REGISTER
3386                                                                      ;UNTIL ALL 18. HAVE BEEN DONE
```

```
3387
3388
3389                                        ;*NOW CLEAR OUT THE CONTROLLER AND DRIVE
3390
3391  013616                          6$:
3392
3393  013616  004737  041524              JSR     PC,@#CLDISK     ;SET R1-RHCS1, R2-RHCS2
3394                                                              ;R3-RHDS1, R4-RHER1
3395                                                              ;GIVE RH-11 INITIALIZE
3396                                                              ;SETUP UNIT NUMBER
3397
3398                                        ;*CHANGE THE REGISTER SNAPSHOT TO EXPECTED VALUES AFTER THE CLEAR
3399
3400
3401  013622  004037  042404              JSR     R0,@#CHREG      ;CHANGE BITS IN SAVED REGISTER
3402  013626  002300                      RHCS1                   ;CHANGE RHCS1 REGISTER
3403
3404  013630  000003                      3                       ;3 BIT/BITS TO BE CHANGED
3405  013632  000000                      0                       ;NEW VALUE OF SC IS 0
3406  013634  100000                      SC                      ;CHANGE SC BIT
3407  013636  000000                      0                       ;NEW VALUE OF TRE IS 0
3408  013640  040000                      TRE                     ;CHANGE TRE BIT
3409  013642  000000                      0                       ;NEW VALUE OF MCPE IS 0
3410  013644  020000                      MCPE                    ;CHANGE MCPE BIT
3411
3412  013646  004037  042404              JSR     R0,@#CHREG      ;CHANGE BITS IN SAVED REGISTER
3413  013652  002322                      RHDS1                   ;CHANGE RHDS1 REGISTER
3414
3415  013654  000002                      2                       ;2 BIT/BITS TO BE CHANGED
3416  013656  000000                      0                       ;NEW VALUE OF ATA IS 0
3417  013660  100000                      ATA                     ;CHANGE ATA BIT
3418  013662  000000                      0                       ;NEW VALUE OF ERR IS 0
3419  013664  040000                      ERR                     ;CHANGE ERR BIT
3420
3421  013666  004037  042404              JSR     R0,@#CHREG      ;CHANGE BITS IN SAVED REGISTER
3422  013672  002276                      RHCS2                   ;CHANGE RHCS2 REGISTER
3423
3424  013674  000001                      1                       ;1 BIT/BITS TO BE CHANGED
3425  013676  000000                      0                       ;NEW VALUE OF NED IS 0
3426  013700  010000                      NED                     ;CHANGE NED BIT
3427
3428  013702  004037  042404              JSR     R0,@#CHREG      ;CHANGE BITS IN SAVED REGISTER
3429  013706  002314                      RHER3                   ;CHANGE RHER3 REGISTER
3430
3431  013710  000001                      1                       ;1 BIT/BITS TO BE CHANGED
3432  013712  000000                      0                       ;NEW VALUE OF ACE IS 0
3433  013714  020000                      ACE                     ;CHANGE ACE BIT
3434
3435  013716  004037  042404              JSR     R0,@#CHREG      ;CHANGE BITS IN SAVED REGISTER
3436  013722  002320                      RHMR                    ;CHANGE RHMR REGISTER
3437
3438  013724  000001                      1                       ;1 BIT/BITS TO BE CHANGED
3439  013726  000000                      0                       ;NEW VALUE OF DMD IS 0
3440  013730  000001                      DMD                     ;CHANGE DMD BIT
3441
3442  013732  043737  004740  004636      BIC     @#ATTENT,@#SAVERE+24 ;UNIT UNDER TEST ATTENTION BIT
```

```
3443
3444
3445                                    ;*TAKE ANOTHER REGISTER SNAPSHOT AND COMPARE RESULTS
3446                                    ;*WITH THE EXPECTED VALUES
3447
3448
3449  013740  004037  042512            JSR     R0,@#COMREG     ;COMPARE SAVED REGISTERS WITH
3450                                                            ;PRESENT VALUE
3451  013744  004612                    SAVERE                  ;GOOD DATA SAVED IN 'SAVERE'
3452  013746  002354                    WC                      ;TEST DATA STARTING FROM 'RHWC'
3453  013750  000022                    18.                     ;18. REGISTERS TO BE COMPARED
3454  013752  013756                    10$                     ;RETURN TO 10$ ON ERROR
3455  013754  013762                    11$                     ;RETURN TO 11$ ON NO ERROR
3456
3457  013756  104100          10$:      ERROR   100             ;ADDRESS PLUG CHANGE CAUSED SOME
3458  013760  000207                    RTS     PC              ;INCORRECT REGISTER RESULT
3459
3460  013762                    11$:
3461
3462                                    ;*(USE NED METHOD TO VERIFY
3463                                    ;*THAT ATTENTION BIT COMES UP IN THE PROPER LOCATION ??)
3464
3465
3466
3467
3468                                    ;*SET 'VV' IN RHDS1 AFTER RESET FROM THE RECALIBRATE
3469                                    ;*CAUSED BY PULLING THE PLUG
3470
3471
3472  013762  013746  002460            MOV     @#PKACK,-(SP)   ;GET READY TO MOVE COMMAND
3473  013766  052716  000001            BIS     #GO,(SP)        ;GET READY TO SET GO
3474                                                            ;WITHOUT INTERRUPT ENABLE
3475  013772  012677  166302            MOV     (SP)+,@RHCS1    ;GO WITH
3476                                                             ;22 IN RHCS1 FOR PACK ACKNOWLEDGE
3477                                                            ;WITH INTERRUPT DISABLED
3478
3479  013776  011100                    MOV     @R1,R0          ;SAVE RHCS1 DURING ABOVE OPERATION
3480  014000  011305                    MOV     @R3,R5          ;SAVE RHDS1 DURING ABOVE OPERATION
3481
3482  014002  104413                    WAT                     ;WAIT FOR VV BIT TO SET
3483  014004  002322                    RHDS1                   ;WAIT FOR RHDS1 REGISTER
3484  014006  000100                    VV                      ;WAIT FOR VV BIT IN RHDS1 REGISTER
3485  014010  000001                    1.                      ;ALLOW 10 MICRO SECONDS
3486  014012  000001                    1.                      ;VV MUST SET BETWEEN
3487                                                            ;OC AND 20 MICRO SECONDS
3488
```

```
3489
3490
3491
3492                                    ;;*****************************************************************
3493                                    ;*TEST 15       CHECK ALL ADDRESS PLUG ADDRESSES
3494
3495                                    ;*      CHECK THAT ALL ADDRESS PLUG NUMBERS FUNCTION PROPERLY
3496
3497                                    ;*      THIS TEST IS DONE FOR MEMOREX RP04'S AND RP06'S ONLY
3498                                    ;*      THIS TEST IS SELECTED BY SW#5 AND THE DEFAULT IS NOT TO DO IT
3499
3500                                    ;;*****************************************************************
3501  014014  000004                    TST15:  SCOPE
3502  014016  012737  000001  001212            MOV     #1,$TIMES       ;;DO 1 ITERATION
3503
3504                                    ;;*****************************************************************
3505  014024  005737  004744                    TST     @#RP06          ;TEST FOR RP06 DRIVE
3506  014030  001005                            BNE     4$              ;IF = 1, OK TO DO THIS TEST
3507  014032  005737  004746                    TST     @#RP05          ;TEST FOR MEMOREX RP04
3508  014036  001002                            BNE     4$              ;IF = 1, OK TO DO THIS TEST
3509                                                                    ;NOT AN RP06 OR MEMOREX RP04 SO
3510                                                                    ;ASSUME AN ISS RP04 AND SKIP TEST
3511  014040  000137  015022                    JMP     TST16   ;       JUMP TO NEXT TEST ------------------)
3512  014044                            4$:
3513                                    ;;*****************************************************************
3514
3515                                            ;*CHECK TO SEE IF THIS TEST HAS BEEN SELECTED WITH SW5
3516
3517  014044  032777  000040  165066             BIT     #SW5,@SWR       ;TEST THE SWITCH
3518  014052  001002                            BNE     5$              ;IF 0, TEST HAS NOT BEEN SELECTED
3519  014054  000137  015022                    JMP     TST16   ;       JUMP TO NEXT TEST ------------------)
3520  014060                            5$:                             ;TEST SELECTED, CONTINUE IT
3521
3522
3523                                            ;*THIS CODE CHECKS TO SEE IF MANUAL INTERVENTION TESTS ARE OK
3524
3525  014060  005737  004724                    TST     @#NOPUSH        ;IS THIS A 220 START ?
3526  014064  001007                            BNE     1$              ;SKIP THIS TEST IF SO
3527  014066  005737  000042                    TST     @#42            ;MONITOR (ACT 11) RETURN ADDRESS ?
3528  014072  001004                            BNE     1$              ;SKIP THIS TEST
3529  014074  005737  001100                    TST     @#$PASS         ;FIRST PASS ?
3530  014100  001001                            BNE     1$              ;SKIP THIS TEST IF NOT
3531  014102  000402                            BR      2$              ;CONTINUE WITH THIS TEST
3532
3533  014104                            1$:
3534  014104  000137  015022                    JMP     TST16   ;       JUMP TO NEXT TEST ----------------)
3535  014110                            2$:
3536  014110  012706  001000                    MOV     #STACK,SP       ;RESET STACK
3537  014114  012737  000015  004604             MOV     #15,@#TSTNM     ;SAVE TEST NUMBER
3538
3539  014122  004737  041524                    JSR     PC,@#CLDISK     ;SET R1-RHCS1, R2-RHCS2
3540                                                                    ;R3-RHDS1, R4-RHER1
3541                                                                    ;GIVE RH-11 INITIALIZE
3542                                                                    ;SETUP UNIT NUMBER
3543  014126  004737  041604                    JSR     PC,@#CHECKT     ;CHECK DVA,RDY,MOL,DPR,DRY,VV = 1
3544                                                                    ;AND THAT NO STATUS BITS ARE STUCK = 1
```

```
3545  014132  104401  067033                  TYPE    ,CPHALT          ;CANNOT CONTINUE TESTING IF ANY OF
3546                                                                   ;THE FIRST SET OF BITS DON'T = 1
3547  014136  000000                          HALT                     ;STOP
3548
3549
3550  014140  104401  014146                  TYPE    ,65$             ;;TYPE ASCIZ STRING
3551  014144  000426                          BR      64$              ;;GET OVER THE ASCIZ
3552                                ;;65$:    .ASCIZ  <15><12>/ALL DISK DRIVES EXCEPTING THE UNIT UNDER/
3553  014222                        64$:
3554  014222  104401  014230                  TYPE    ,67$             ;;TYPE ASCIZ STRING
3555  014226  000417                          BR      66$              ;;GET OVER THE ASCIZ
3556                                ;;67$:    .ASCIZ  <15><12>/TEST MUST BE POWERED DOWN/<15><12>
3557  014266                        66$:
3558
3559
3560
3561  014266  013737  004716  004752          MOV     @#UNIT,@#INUNIT ;MAKE THE INITIAL UNIT NO. = "UNIT"
3562
```

```
                                        ;*CHANGE ADDRESS PLUG ON THE UNIT UNDER TEST
014274                          6$:
014274  104401  014302                TYPE    ,69$            ;;TYPE ASCIZ STRING
014300  000420                        BR      68$             ;;GET OVER THE ASCIZ
                                ;;69$:  .ASCIZ  <15><12>/REMOVE ADDRESS PLUG ON DRIVE/
014342                          68$:
014342  013746  004716                MOV     @#UNIT,-(SP)    ;GET THE UNIT UNDER TEST
014346  104405                        TYPDS                   ;TYPE IT OUT
014350  104401  014356                TYPE    ,71$            ;;TYPE ASCIZ STRING
014354  000421                        BR      70$             ;;GET OVER THE ASCIZ
                                ;;71$:  .ASCIZ  /       REPLACE WITH NEXT HIGHER ADDRESS/
014420                          70$:
014420  104401  014426                TYPE    ,73$            ;;TYPE ASCIZ STRING
014424  000417                        BR      72$             ;;GET OVER THE ASCIZ
                                ;;73$:  .ASCIZ  <15><12>/PLUG - THEN PRESS CONTINUE/
014464                          72$:
014464  000000                        HALT
                                        ;*HOUSEKEEPING
014466  005037  047342                CLR     @#PRITEM        ;CLEAR THE PREVIOUS ERROR NUMBER
014472  005237  004716                INC     @#UNIT          ;ADD ONE TO THE UNIT NO.
014476  042737  177770  004716        BIC     #^C7,@#UNIT     ;TRUNCATE TO LOW ORDER 3 BITS
014504  004737  041524                JSR     PC,@#CLDISK     ;SET R1-RHCS1, R2-RHCS2
                                                            ;R3-RHDS1, R4-RHER1
                                                            ;GIVE RH-11 INITIALIZE
                                                            ;SETUP UNIT NUMBER
                                        ;*ATTEMPT TO ADDRESS THE NEW UNIT NUMBER
014510  017700  165570                MOV     @RHDST,R0       ;ATTEMPT TO ADDRESS THE NEW DRIVE NO.
014514  004737  043436                JSR     PC,@#PUTREG     ;TAKE REG. SNAPSHOT IN CASE OF ERROR
014520  032737  010000  002360        BIT     #NED,@#CS2      ;TEST FOR NON EXISTENT DRIVE
014526  001423                        BEQ     7$              ;CONTINUE IF 'NED' IS NOT SET - DRIVE
                                                            ;SHOULD BE EXISTENT ON THE BUSS
014530  104102                        ERROR   102             ;UNIT NOT AVAILABLE AFTER ADDRESS
                                                            ;PLUG REPLACED
014532  104401  014540                TYPE    ,75$            ;;TYPE ASCIZ STRING
014536  000407                        BR      74$             ;;GET OVER THE ASCIZ
                                ;;75$:  .ASCIZ  <15><12>/UNIT NUMBER/
014556                          74$:
014556  013746  004716                MOV     @#UNIT,-(SP)    ;GET THE BAD UNIT NUMBER
014562  104405                        TYPDS                   ;TYPE IT OUT
014564  104401  014572                TYPE    ,77$            ;;TYPE ASCIZ STRING
014570  000402                        BR      76$             ;;GET OVER THE ASCIZ
                                ;;77$:  .ASCIZ  <15><12>/ /
014576                          76$:
```

```
3616
3617                                        ;*CHECK IF ALL UNIT NUMBERS HAVE BEEN TRIED
3618
3619  014576  023737  004716  004752  7$:   CMP     @#UNIT,@#INUNIT ;HAVE WE INCREMENTED BACK TO THE
3620                                                                ;ORIGINAL UNIT NO. YET ?
3621  014604  001233                        BNE     6$              ;NO..DO NEXT ADDRESS PLUG
3622                                                                ;YES..CONTINUE WITH TESTS
3623
3624
3625
3626                                        ;*SET 'VV'IN RHDS1 AFTER RESET FROM THE RECALIBRATE
3627                                        ;*CAUSED BY PULLING THE ADDRESS PLUGS OUT
3628
3629
3630  014606  013746  002460                MOV     @#PKACK,-(SP)   ;GET READY TO MOVE COMMAND
3631  014612  052716  000001                BIS     #GO,(SP)        ;GET READY TO SET GO
3632                                                                ;WITHOUT INTERRUPT ENABLE
3633  014616  012677  165456                MOV     (SP)+,@RHCS1    ;GO WITH
3634                                                                 ;22 IN RHCS1 FOR PACK ACKNOWLEDGE
3635                                                                ;WITH INTERRUPT DISABLED
3636
3637  014622  011100                        MOV     @R1,R0          ;SAVE RHCS1 DURING ABOVE OPERATION
3638  014624  011305                        MOV     @R3,R5          ;SAVE RHDS1 DURING ABOVE OPERATION
3639
3640  014626  104413                        WAT                     ;WAIT FOR VV BIT TO SET
3641  014630  002322                        RHDS1                   ;WAIT FOR RHDS1 REGISTER
3642  014632  000100                        VV                      ;WAIT FOR VV BIT IN RHDS1 REGISTER
3643  014634  000001                        1.                      ;ALLOW 10 MICRO SECONDS
3644  014636  000001                        1.                      ;VV MUST SET BETWEEN
3645                                                                ;00 AND 20 MICRO SECONDS
3646
3647  014640  104401  014646                TYPE    ,82$            ;;TYPE ASCIZ STRING
3648  014644  000436                        BR      81$             ;;GET OVER THE ASCIZ
3649                                ;;82$:   .ASCIZ  <15><12><15><12>/ALL DISK DRIVES WHICH WERE POWERED UP WHEN THE PROGRAM/
3650  014742                        81$:
3651  014742  104401  014750                TYPE    ,84$            ;;TYPE ASCIZ STRING
3652  014746  000425                        BR      83$             ;;GET OVER THE ASCIZ
3653                                ;;84$:   .ASCIZ  <15><12>/WAS STARTED MUST BE POWERED UP AGAIN/<15><12>
3654  015022                        83$:
```

```
3655
3656
3657                                    .SBTTL
3658                                    .SBTTL  **DRIVE COMMAND TESTS**
3659                                    .SBTTL
3660
3661
3662
3663
3664
3665
3666                                    ;;*****************************************************************
3667                                    ;*TEST 16       NO OPERATION FUNCTION TEST
3668                                    ;*      ALL POSSIBLE REGISTERS ARE CLEARED THEN A''NOP''=0
3669                                    ;*      IS GIVEN NO CHANGE SHOULD HAPPEN
3670                                    ;*      ALL POSSIBLE REGISTERS ARE FILLED WITH ONES THEN A ''NOP''
3671                                    ;*      IS GIVEN NO CHANGE SHOULD HAPPEN
3672
3673                                    ;;*****************************************************************
3674  015022  000004                   TST16:  SCOPE
3675  015024  012706  001000                   MOV     #STACK,SP       ;RESET STACK
3676  015030  012737  000016  004604           MOV     #16,@#TSTNM     ;SAVE TEST NUMBER
3677
3678  015036  004737  041524                   JSR     PC,@#CLDISK     ;SET R1-RHCS1, R2-RHCS2
3679                                                                    ;R3-RHDS1, R4-RHER1
3680                                                                    ;GIVE RH-11 INITIALIZE
3681                                                                    ;SETUP UNIT NUMBER
3682  015042  004737  041604                   JSR     PC,@#CHECKT     ;CHECK DVA,RDY,MOL,DPR,DRY,VV = 1
3683                                                                    ;AND THAT NO STATUS BITS ARE STUCK = 1
3684  015046  104401  067033                   TYPE    ,CPHALT         ;CANNOT CONTINUE TESTING IF ANY OF
3685                                                                    ;THE FIRST SET OF BITS DON'T = 1
3686  015052  000000                           HALT                    ;STOP
3687
3688  015054  013777  002422  165216           MOV     @#NOPERA,@RHCS1 ;GET READY FOR NOPERA
3689                                                                    ;NO OPERATION WITH 0 IN RHCS1
3690
3691
3692                                    ;*NOW SAVE REGISTERS FOR COMPARISON AFTER NO OPERATION
3693
3694  015062  004037  041672                   JSR     R0,@#SAVER      ;SAVE REGISTERS
3695  015066  002272                           RHWC                    ;RHWC IS THE FIRST REGISTER SAVED
3696  015070  004612                           SAVERE                  ;STARTING ADDRESS OF WHERE
3697                                                                    ;THE REGISTERS ARE SAVED
3698  015072  000022                           18.                     ;NUMBER OF REGISTERS
3699                                                                    ;SAVED = 18.
3700
3701  015074  013777  004606  165164           MOV     @#RP4VEC,@RPVEC ;SET RP04 VECTOR ADDRESS
3702                                                                    ;TO 'TIME1' IF P-CLOCK IS PRESENT
3703                                                                    ;OR TO 'TIME2' IF P-CLOCK IS NOT PRESENT
3704                                                                    ;'TIME' WILL ONLY SAVE
3705                                                                    ;CURRENT CYLINDER ADDRESS
3706                                                                    ;AND LOOK AHEAD REGISTERS
3707
3708
3709  015102  013746  002422                   MOV     @#NOPERA,-(SP)  ;GET READY TO MOVE COMMAND
3710  015106  052716  000001                   BIS     #GO,(SP)        ;GET READY TO SET GO
```

```
3711                                                    ;WITHOUT INTERRUPT ENABLE
3712  015112  012677  165162          MOV     (SP)+,@RHCS1    ;GO WITH
3713                                                    ;0 IN RHCS1 FOR NO-OPERATION
3714                                                    ;WITH INTERRUPT DISABLED
3715
3716
3717
3718  015116  104413                  WAT                     ;WAIT FOR RDY BIT TO SET
3719  015120  002300                  RHCS1                   ;WAIT FOR RHCS1 REGISTER
3720  015122  000200                  RDY                     ;WAIT FOR RDY BIT IN RHCS1 REGISTER
3721  015124  000001                  1.                      ;ALLOW 10 MICRO SECONDS
3722  015126  000001                  1.                      ;RDY MUST SET BETWEEN
3723                                                    ;00 AND 20 MICRO SECONDS
3724
3725                                  ;*NOW COMPARE REGISTERS BEFORE NO-OP COMMAND
3726                                  ;*WITH AFTER NO-OP COMMAND
3727
3728  015130  004037  042512          JSR     R0,@#COMREG     ;COMPARE SAVED REGISTERS WITH
3729                                                    ;PRESENT VALUE
3730  015134  004612                  SAVERE                  ;GOOD DATA SAVED IN 'SAVERE'
3731  015136  002354                  WC                      ;TEST DATA STARTING FROM 'RHWC'
3732  015140  000022                  18.                     ;18. REGISTERS TO BE COMPARED
3733  015142  015146                  1$                      ;RETURN TO 1$ ON ERROR
3734  015144  015152                  2$                      ;RETURN TO 2$ ON NO ERROR
3735
3736  015146  104016          1$:     ERROR   16              ;GIVING A NO-OP COMMAND
3737  015150  000207                  RTS     PC              ;CAUSED AN ERROR
3738                                                    ;NO REGISTERS SHOULD CHANGE
3739                                                    ;GOOD DATA GIVES REGISTER
3740                                                    ;CONTENTS BEFORE COMMAND
3741                                                    ;RECEIVED DATA GIVES REGISTER
3742                                                    ;CONTENTS AFTER COMMAND
3743
```

```
3744
3745                                        ;*NOW REPEAT TEST BY MOVING IN ALL POSSIBLE ONES
3746
3747  015152                          2$:
3748
3749  015152  004737  041524               JSR     PC,@#CLDISK     ;SET R1-RHCS1, R2-RHCS2
3750                                                                 ;R3-RHDS1, R4-RHER1
3751                                                                 ;GIVE RH-11 INITIALIZE
3752                                                                 ;SETUP UNIT NUMBER
3753  015156  012700  002272               MOV     #RHWC,R0        ;ADDR. OF ADDR OF RHWC IN R0
3754  015162  012730  177777               MOV     #177777,@(R0)+  ;LOAD 177777 INTO RHWC
3755
3756  015166  012730  177777               MOV     #177777,@(R0)+  ;LOAD 177777 INTO RHBA
3757
3758  015172  052730  043010               BIS     #43010,@(R0)+   ;LOAD 43010 INTO RHCS2
3759  015176  012730  001400               MOV     #1400,@(R0)+    ;LOAD 1400 INTO RHCS1
3760
3761  015202  012730  000000               MOV     #0,@(R0)+       ;LOAD 0 INTO RHER1
3762
3763  015206  012730  177777               MOV     #177777,@(R0)+  ;LOAD 177777 INTO RHDST
3764
3765  015212  012730  000000               MOV     #0,@(R0)+       ;LOAD 0 INTO RHER2
3766
3767  015216  012730  177777               MOV     #177777,@(R0)+  ;LOAD 177777 INTO RHOF
3768
3769  015222  012730  177777               MOV     #177777,@(R0)+  ;LOAD 177777 INTO RHCA
3770
3771  015226  012730  000000               MOV     #0,@(R0)+       ;LOAD 0 INTO RHER3
3772
3773                                        ;*NOW SET BITS IN RHAS FOR ALL DRIVES PRESENT
3774
3775  015232  010046                     MOV     R0,-(SP)        ;;PUSH R0 ON STACK
3776  015234  011446                     MOV     @R4,-(SP)       ;SAVE RHER1 TO REINSTATE LATER
3777  015236  011246                     MOV     @R2,-(SP)       ;SAVE RHCS2 TO BE REINSTATED
3778                                                              ;AFTER ALL ATA BITS HAVE BEEN SET
3779  015240  013700  004742               MOV     @#TOTALAT,R0    ;GET DRIVES PRESENT
3780  015244  005012                     CLR     @R2             ;CLEAR RHCS2 AND CARRY
3781  015246  012705  000010               MOV     #8.,R5          ;COUNTER
3782  015252  006000                 87$:  ROR     R0              ;GET BIT INTO CARRY
3783  015254  103002                     BCC     88$             ;BRANCH IF NO UNIT ON THIS BIT
3784  015256  012714  177777               MOV     #-1,@R4         ;MOVE INTO ERROR REGISTER
3785                                                              ;TO SET ATA BIT
3786  015262  005212                 88$:  INC     @R2             ;INCREMENT RHCS2 TO NEXT UNIT
3787  015264  005305                     DEC     R5              ;COUNT
3788  015266  001371                     BNE     87$             ;BRANCH IF 8 NOT DONE
3789  015270  012612                     MOV     (SP)+,@R2       ;REINSTATE RHCS2
3790  015272  012614                     MOV     (SP)+,@R4       ;REINSTATE RHER1
3791  015274  012600                     MOV     (SP)+,R0        ;;POP STACK INTO R0
3792  015276  005720                     TST     (R0)+           ;GET OVER PHAS IN R0
3793
3794
3795  015300  012730  177776               MOV     #177776,@(R0)+  ;LOAD 177776 INTO RHMR
3796
3797
3798  015304  004737  041604               JSR     PC,@#CHECKT     ;CHECK DVA,RDY,MOL,DPR,DRY,VV = 1
3799                                                                 ;AND THAT NO STATUS BITS ARE STUCK = 1
```

```
3800  015310  104401  067033                  TYPE    ,CPHALT          ;CANNOT CONTINUE TESTING IF ANY OF
3801                                                                     ;THE FIRST SET OF BITS DON'T = 1
3802  015314  000000                          HALT                     ;STOP
3803  015316  013777  002422  164754          MOV     @#NOPERA,@RHCS1  ;GET READY FOR NOPERA
3804                                                                     ;NO OPERATION WITH 0 IN RHCS1
3805
3806
3807                                          ;*NOW SAVE REGISTERS FOR COMPARISON AFTER A NO-OP
3808
3809  015324  004037  041672                  JSR     R0,@#SAVER       ;SAVE REGISTERS
3810  015330  002272                          RHWC                     ;RHWC IS THE FIRST REGISTER SAVED
3811  015332  004612                          SAVERE                   ;STARTING ADDRESS OF WHERE
3812                                                                     ;THE REGISTERS ARE SAVED
3813  015334  000022                          18.                      ;NUMBER OF REGISTERS
3814                                                                     ;SAVED = 18.
3815
3816  015336  013777  004606  164722          MOV     @#RP4VEC,@RPVEC  ;SET RP04 VECTOR ADDRESS
3817                                                                     ;TO 'TIME1' IF P-CLOCK IS PRESENT
3818                                                                     ;OR TO 'TIME2' IF P-CLOCK IS NOT PRESENT
3819                                                                     ;'TIME' WILL ONLY SAVE
3820                                                                     ;CURRENT CYLINDER ADDRESS
3821                                                                     ;AND LOOK AHEAD REGISTERS
3822
3823
3824  015344  013746  002422                  MOV     @#NOPERA,-(SP)   ;GET READY TO MOVE COMMAND
3825  015350  052716  000001                  BIS     #GO,(SP)         ;GET READY TO SET GO
3826                                                                     ;WITHOUT INTERRUPT ENABLE
3827  015354  012677  164720                  MOV     (SP)+,@RHCS1     ;GO WITH
3828                                                                     ;0 IN RHCS1 FOR NO-OPERATION
3829                                                                     ;WITH INTERRUPT DISABLED
3830
3831
3832
3833  015360  104413                          WAT                      ;WAIT FOR RDY BIT TO SET
3834  015362  002300                          RHCS1                    ;WAIT FOR RHCS1 REGISTER
3835  015364  000200                          RDY                      ;WAIT FOR RDY BIT IN RHCS1 REGISTER
3836  015366  000001                          1.                       ;ALLOW 10 MICRO SECONDS
3837  015370  000001                          1.                       ;RDY MUST SET BETWEEN
3838                                                                     ;00 AND 20 MICRO SECONDS
3839
3840                                          ;*CHANGE REGISTERS TO EXPECTED VALUES
3841
3842  015372  005737  004750                  TST     @#RH70           ;RUNNING ON AN RH70 ?
3843  015376  001406                          BEQ     5$               ;IF NOT, SKIP NEXT
3844  015400  005737  004722                  TST     @#NUNIT          ;TESTING MORE THAN ONE DRIVE ?
3845  015404  001003                          BNE     5$               ;SKIP NEXT IF SO
3846  015406  042737  100000  004620          BIC     #SC,@#SAVERE+6   ;CLEAR 'SC' IN RHCS1
3847
3848  015414                          5$:
3849
3850  015414  043737  004740  004636          BIC     @#ATTENT,@#SAVERE+24     ;CLEAR APPROPIATE ATA BITS
3851                                                                           ;FOR WORKING DRIVE IN SAVED RHAS
3852
3853  015422  004037  042404                  JSR     R0,@#CHREG       ;CHANGE BITS IN SAVED REGISTER
3854  015426  002322                          RHDS1                    ;CHANGE RHDS1 REGISTER
3855
```

```
3856  015430  000001                    1                       ;1 BIT/BITS TO BE CHANGED
3857  015432  000000                    0                       ;NEW VALUE OF ATA IS 0
3858  015434  100000                    ATA                     ;CHANGE ATA BIT
3859
3860
3861                                    ;*NOW COMPARE REGISTERS BEFORE NO-OP WITH
3862                                    ;*AFTER NO-OP COMMAND
3863
3864
3865  015436  004037  042512            JSR     R0,@#COMREG     ;COMPARE SAVED REGISTERS WITH
3866                                                            ;PRESENT VALUE
3867  015442  004612                    SAVERE                  ;GOOD DATA SAVED IN 'SAVERE'
3868  015444  002354                    WC                      ;TEST DATA STARTING FROM 'RHWC'
3869  015446  000022                    18.                     ;18. REGISTERS TO BE COMPARED
3870  015450  015454                    3$                      ;RETURN TO 3$ ON ERROR
3871  015452  015460                    4$                      ;RETURN TO 4$ ON NO ERROR
3872
3873  015454  104016            3$:     ERROR   16              ;GIVING A NO-OP COMMAND
3874  015456  000207                    RTS     PC              ;CAUSED AN ERROR
3875                                                            ;NO REGISTERS SHOULD CHANGE
3876                                                            ;GOOD DATA GIVES REGISTER
3877                                                            ;CONTENTS BEFORE COMMAND
3878                                                            ;RECEIVED DATA GIVES REGISTER
3879                                                            ;CONTENTS AFTER COMMAND
3880  015460            4$:
3881
3882
```

```
                                ;;*****************************************************************
                                ;*TEST 17       DRIVE CLEAR
                                ;*      ALL WRITE BITS OF ALL REGISTERS ARE FILLED
                                ;*      WITH ONES EXCEPT GO,CLR,IE,PAT,MCPE,UPE
                                ;*      THEN A DRIVE CLEAR IS GIVEN
                                ;*      THEN ALL REGISTERS ARE CHECKED TO HAVE APPROPIAVE VALUE
                                ;;*****************************************************************
015460  000004                TST17:  SCOPE
015462  012706  001000                MOV     #STACK,SP       ;RESET STACK
015466  012737  000017  004604        MOV     #17,@#TSTNM     ;SAVE TEST NUMBER

015474  004737  041524                JSR     PC,@#CLDISK     ;SET R1-RHCS1, R2-RHCS2
                                                              ;R3-RHDS1, R4-RHER1
                                                              ;GIVE RH-11 INITIALIZE
                                                              ;SETUP UNIT NUMBER
015500  004737  041604                JSR     PC,@#CHECKT     ;CHECK DVA,RDY,MOL,DPR,DRY,VV = 1
                                                              ;AND THAT NO STATUS BITS ARE STUCK = 1
015504  104401  067033                TYPE    ,CPHALT         ;CANNOT CONTINUE TESTING IF ANY OF
                                                              ;THE FIRST SET OF BITS DON'T = 1
015510  000000                        HALT                    ;STOP

                                ;*WRITE ALL WRITABLE REGISTER BITS

015512  012700  002272                MOV     #RHWC,R0        ;ADDR. OF ADDR. OF RHWC IN R0
015516  012730  177777                MOV     #177777,@(R0)+  ;LOAD 177777 INTO RHWC

015522  012730  177776                MOV     #177776,@(R0)+  ;LOAD 177776 INTO RHBA

015526  052730  043010                BIS     #43010,@(R0)+   ;LOAD 43010 INTO RHCS2
015532  012730  001400                MOV     #1400,@(R0)+    ;LOAD 1400 INTO RHCS1

015536  012730  000000                MOV     #0,@(R0)+       ;LOAD 0 INTO RHER1

015542  012730  177777                MOV     #177777,@(R0)+  ;LOAD 177777 INTO RHDST

015546  012730  000000                MOV     #0,@(R0)+       ;LOAD 0 INTO RHER2

015552  012730  177777                MOV     #177777,@(R0)+  ;LOAD 177777 INTO RHOF

015556  012730  177777                MOV     #177777,@(R0)+  ;LOAD 177777 INTO RHCA

015562  012730  000000                MOV     #0,@(R0)+       ;LOAD 0 INTO RHER3

                                ;*NOW SET BITS IN RHAS FOR ALL DRIVES PRESENT

015566  010046                        MOV     R0,-(SP)        ;;PUSH R0 ON STACK
015570  011446                        MOV     @R4,-(SP)       ;SAVE RHER1 TO REINSTATE LATER
015572  011246                        MOV     @R2,-(SP)       ;SAVE RHCS2 TO BE REINSTATED
                                                              ;AFTER ALL ATA BITS HAVE BEEN SET
015574  013700  004742                MOV     @#TOTALAT,R0    ;GET DRIVES PRESENT
015600  005012                        CLR     @R2             ;CLEAR RHCS2 AND CARRY
015602  012705  000010                MOV     #8.,R5          ;COUNTER
015606  006000                84$:    ROR     R0              ;GET BIT INTO CARRY
015610  103002                        BCC     85$             ;BRANCH IF NO UNIT ON THIS BIT
015612  012714  177777                MOV     #-1,@R4         ;MOVE INTO ERROR REGISTER
                                                              ;TO SET ATA BIT
```

```
3939  015616  005212                    85$:    INC     @R2             ;INCREMENT RHCS2 TO NEXT UNIT
3940  015620  005305                            DEC     R5              ;COUNT
3941  015622  001371                            BNE     84$             ;BRANCH IF 8 NOT DONE
3942  015624  012612                            MOV     (SP)+,@R2       ;REINSTATE RHCS2
3943  015626  012614                            MOV     (SP)+,@R4       ;REINSTATE RHER1
3944  015630  012600                            MOV     (SP)+,R0        ;;POP STACK INTO R0
3945  015632  005720                            TST     (R0)+           ;GET OVER PHAS IN R0
3946
3947
3948  015634  012730  177776                    MOV     #177776,@(R0)+  ;LOAD 177776 INTO RHMR
3949
3950
3951  015640  017737  164470  004654            MOV     @RHCC,@#SAVERE+42 ;SAVE RHCC IN SAVERE TABLE
3952  015646  013777  004606  164412            MOV     @#RP4VEC,@RPVEC ;SET RP04 VECTOR ADDRESS
3953                                                                    ;TO 'TIME1' IF P-CLOCK IS PRESENT
3954                                                                    ;OR TO 'TIME2' IF P-CLOCK IS NOT PRESENT
3955                                                                    ;'TIME' WILL ONLY SAVE
3956                                                                    ;CURRENT CYLINDER ADDRESS
3957                                                                    ;AND LOOK AHEAD REGISTERS
3958
3959
3960  015654  013746  002430                    MOV     @#DCLEAR,-(SP)  ;GET READY TO MOVE COMMAND
3961  015660  052716  000001                    BIS     #GO,(SP)        ;GET READY TO SET GO
3962                                                                    ;WITHOUT INTERRUPT ENABLE
3963  015664  012677  164410                    MOV     (SP)+,@RHCS1    ;GO WITH
3964                                                                     ;10 IN RHCS1 FOR DRIVE CLEAR
3965                                                                    ;WITH INTERRUPT DISABLED
3966
3967
3968
3969  015670  104413                            WAT                     ;WAIT FOR RDY BIT TO SET
3970  015672  002300                            RHCS1                   ;WAIT FOR RHCS1 REGISTER
3971  015674  000200                            RDY                     ;WAIT FOR RDY BIT IN RHCS1 REGISTER
3972  015676  000001                            1.                      ;ALLOW 10 MICRO SECONDS
3973  015700  000001                            1.                      ;RDY MUST SET BETWEEN
3974                                                                    ;00 AND 20 MICRO SECONDS
3975
3976                                            ;*NOW LOAD 'SAVERE' REGISTER SNAPSHOT WITH EXPECTED VALUES
3977
3978  015702  004037  041426                    JSR     R0,@#FILLRE     ;MOV 177777 INTO SAVED RHWC
3979  015706  002272                            RHWC                    ;SAVED REGISTER TO CHANGE
3980  015710  177777                            177777                  ;DATA
3981  015712  004037  041426                    JSR     R0,@#FILLRE     ;MOV 177776 INTO SAVED RHBA
3982  015716  002274                            RHBA                    ;SAVED REGISTER TO CHANGE
3983  015720  177776                            177776                  ;DATA
3984  015722  005037  004616                    CLR     @#SAVERE+4      ;CLEAR LOCATION FOR RHCS2
3985  015726  053737  004716  004616            BIS     @#UNIT,@#SAVERE+4;PUT UNIT # BACK IN THE SAVED RHCS2
3986
3987  015734  005737  004750                    TST     @#RH70          ;RUNNING ON AN RH70 CONTROLLER ?
3988  015740  001021                            BNE     8$              ;IF SO SKIP NEXT RH11 CODE
3989
3990
3991  015742  004037  042404                    JSR     R0,@#CHREG      ;CHANGE BITS IN SAVED REGISTER
3992  015746  002276                            RHCS2                   ;CHANGE RHCS2 REGISTER
3993
3994  015750  000003                            3                       ;3 BIT/BITS TO BE CHANGED
```

```
 3995  015752  000001                          1                       ;NEW VALUE OF IR IS 1
 3996  015754  000100                          IR                      ;CHANGE IR BIT
 3997  015756  000001                          1                       ;NEW VALUE OF BAI IS 1
 3998  015760  000010                          BAI                     ;CHANGE BAI BIT
 3999  015762  000001                          1                       ;NEW VALUE OF MXF IS 1
 4000  015764  001000                          MXF                     ;CHANGE MXF BIT
 4001
 4002  015766  004037  042404                  JSR     R0,@#CHREG      ;CHANGE BITS IN SAVED REGISTER
 4003  015772  002300                          RHCS1                   ;CHANGE RHCS1 REGISTER
 4004
 4005  015774  000001                          1                       ;1 BIT/BITS TO BE CHANGED
 4006  015776  000001                          1                       ;NEW VALUE OF SC IS 1
 4007  016000  100000                          SC                      ;CHANGE SC BIT
 4008  016002  000416                          BR      9$              ;SKIP NEXT RH70 CODE
 4009
 4010  016004                          8$:
 4011
 4012  016004  004037  042404                  JSR     R0,@#CHREG      ;CHANGE BITS IN SAVED REGISTER
 4013  016010  002276                          RHCS2                   ;CHANGE RHCS2 REGISTER
 4014
 4015  016012  000002                          2                       ;2 BIT/BITS TO BE CHANGED
 4016  016014  000001                          1                       ;NEW VALUE OF IR IS 1
 4017  016016  000100                          IR                      ;CHANGE IR BIT
 4018  016020  000001                          1                       ;NEW VALUE OF BAI IS 1
 4019  016022  000010                          BAI                     ;CHANGE BAI BIT
 4020  016024  005737  004722                  TST     @#NUNIT         ;TESTING MORE THAN ONE DRIVE ?
 4021  016030  001003                          BNE     9$              ;SKIP NEXT IF SO
 4022  016032  042737  100000  004620          BIC     #SC,@#SAVERE+6  ;CLEAR 'SC' IF NOT
 4023  016040                          9$:
 4024  016040  004037  041426                  JSR     R0,@#FILLRE     ;MOV 0 INTO SAVED RHER1
 4025  016044  002302                          RHER1                   ;SAVED REGISTER TO CHANGE
 4026  016046  000000                          0                       ;DATA
 4027  016050  004037  041426                  JSR     R0,@#FILLRE     ;MOV 17437 INTO SAVED RHDST
 4028  016054  002304                          RHDST                   ;SAVED REGISTER TO CHANGE
 4029  016056  017437                          17437                   ;DATA
 4030  016060  004037  041426                  JSR     R0,@#FILLRE     ;MOV 0 INTO SAVED RHER2
 4031  016064  002306                          RHER2                   ;SAVED REGISTER TO CHANGE
 4032  016066  000000                          0                       ;DATA
 4033  016070  004037  041426                  JSR     R0,@#FILLRE     ;MOV 116000 INTO SAVED RHOF
 4034  016074  002310                          RHOF                    ;SAVED REGISTER TO CHANGE
 4035  016076  116000                          116000                  ;DATA
 4036  016100  004037  041426                  JSR     R0,@#FILLRE     ;MOV 1777 INTO SAVED RHCA
 4037  016104  002312                          RHCA                    ;SAVED REGISTER TO CHANGE
 4038  016106  001777                          1777                    ;DATA
 4039  016110  004037  041426                  JSR     R0,@#FILLRE     ;MOV 0 INTO SAVED RHER3
 4040  016114  002314                          RHER3                   ;SAVED REGISTER TO CHANGE
 4041  016116  000000                          0                       ;DATA
 4042  016120  013746  004742                  MOV     @#TOTALAT,-(SP) ;GET ALL BITS OF DRIVE & PRESENT
 4043                                          ;IN RHAS
 4044  016124  043716  004740                  BIC     @#ATTENT,(SP)   ;CLEAR WORKING DRIVE BIT
 4045  016130  012637  004636                  MOV     (SP)+,@#SAVERE+24 ;MOVE THIS INTO RHAS POSITION
 4046  016134  004037  041426                  JSR     R0,@#FILLRE     ;MOV 400 INTO SAVED RHMR
 4047  016140  002320                          RHMR                    ;SAVED REGISTER TO CHANGE
 4048  016142  000400                          400                     ;DATA
 4049
 4050  016144                          3$:
```

```
4051  016144  004037  041426                    JSR     R0,@#FILLRE     ;MOV 10700 INTO SAVED RHDS1
4052  016150  002322                            RHDS1                   ;SAVED REGISTER TO CHANGE
4053  016152  010700                            10700                   ;DATA
4054
4055  016154  013737  002406  004644  4$:       MOV     @#DT,@#SAVERE+32 ;MOVE DRIVE TYPE INTO RHDT
4056                                                                    ;POSITION OF SAVRE TABLE
4057  016162  013737  002410  004646            MOV     @#SN,@#SAVERE+34 ;MOVE SERIAL NUMBER INTO RHSN
4058                                                                    ;POSITION OF SAVERE TABLE
4059
4060  016170  004037  041426                    JSR     R0,@#FILLRE     ;MOV 0 INTO SAVED RHEC1
4061  016174  002330                            RHEC1                   ;SAVED REGISTER TO CHANGE
4062  016176  000000                            0                       ;DATA
4063  016200  004037  041426                    JSR     R0,@#FILLRE     ;MOV 0 INTO SAVED RHEC2
4064  016204  002332                            RHEC2                   ;SAVED REGISTER TO CHANGE
4065  016206  000000                            0                       ;DATA
4066
4067  016210  004037  042404                    JSR     R0,@#CHREG      ;CHANGE BITS IN SAVED REGISTER
4068  016214  002300                            RHCS1                   ;CHANGE RHCS1 REGISTER
4069
4070  016216  000001                            1                       ;1 BIT/BITS TO BE CHANGED
4071  016220  000001                            1                       ;NEW VALUE OF PAR IS 1
4072  016222  000010                            PAR                     ;CHANGE PAR BIT
4073
4074                                            ;*NOW THAT SAVERE TABLE HAS BEEN LOADED WITH
4075                                            ;*EXPECTED VALUES, THE REGISTERS WILL BE COMPARED
4076                                            ;*WITH SAVERE TABLE
4077
4078
4079  016224  004037  042512                    JSR     R0,@#COMREG     ;COMPARE SAVED REGISTERS WITH
4080                                                                    ;PRESENT VALUE
4081  016230  004612                            SAVERE                  ;GOOD DATA SAVED IN 'SAVERE'
4082  016232  002354                            WC                      ;TEST DATA STARTING FROM 'RHWC'
4083  016234  000022                            18.                     ;18. REGISTERS TO BE COMPARED
4084  016236  016242                            5$                      ;RETURN TO 5$ ON ERROR
4085  016240  016246                            6$                      ;RETURN TO 6$ ON NO ERROR
4086
4087  016242  104017                 5$:        ERROR   17              ;DRIVE CLEAR COMMAND
4088  016244  000207                            RTS     PC              ;GAVE AN ERROR
4089                                                                    ;GOOD DATA HAS WHAT SHOULD
4090                                                                    ;BE IN REGISTER AFTER A
4091                                                                    ;DRIVE CLEAR
4092                                                                    ;RECEIVED DATA HAS WHAT
4093                                                                    ;THE REGISTER ACTUALLY
4094                                                                    ;CONTAINED
4095  016246                       6$:
4096
```

```
4097
4098                            ;;*****************************************************************
4099                            ;*TEST 20       READ-IN-PRESET
4100                            ;*      IF STARTING ADDRESS 220 IS USED THIS TEST WILL NOT BE PERFORMED
4101                            ;*
4102                            ;*      IF THE PROGRAM WORKS UNDER ACT-11 MONITOR
4103                            ;*      THEN THIS TEST IS NOT PERFORMED
4104                            ;*
4105                            ;*      IF NO ACT-11 MONITOR IS PRESENT THEN
4106                            ;*      THIS TEST IS PERFORMED ONLY ON THE FIRST PASS
4107                            ;*      ON SUBSEQUENT PASSES THIS TEST IS NOT DONE
4108                            ;*
4109                            ;*
4110                            ;*      THIS TESTS THE READ-IN-PRESET COMMAND
4111                            ;*      FIRST THE DRIVE IS POWERED DOWN AND UP IN ORDER TO
4112                            ;*      RESET VV-BIT #6 IN RHDS1
4113                            ;*      THEN ALL WRITE BITS OF ALL REGISTERS ARE FILLED
4114                            ;*      WITH ONES EXCEPT GO,CLR,IE,PAT,MCPE,UPE
4115                            ;*      ATA FOR DRIVE UNDER TEST IS MADE = 0
4116                            ;*
4117                            ;*
4118                            ;*      THEN A READ-IN-PRESET COMMAND = 20 IS GIVEN
4119                            ;*      THEN ALL REGISTERS ARE TESTED
4120                            ;*      THE FOLLOWING SHOULD BE THE REGISTER CONTENTS
4121                            ;*      RHCA = 0, RHDST = 0, RHOF SHOULD HAVE
4122                            ;*      FMT22 = 0, ECI = 0, HCI = 0, RHDS1 SHOULD HAVE VV = 1
4123                            ;*      ALL OTHER REGISTERS SHOULD BE UNCHANGED
4124
4125                            ;;*****************************************************************
4126  016246  000004            TST20:  SCOPE
4127  016250  012706  001000            MOV     #STACK,SP       ;RESET STACK
4128  016254  012737  000020  004604    MOV     #20,@#TSTNM     ;SAVE TEST NUMBER
4129
4130  016262  004737  041524            JSR     PC,@#CLDISK     ;SET R1-RHCS1, R2-RHCS2
4131                                                            ;R3-RHDS1, R4-RHER1
4132                                                            ;GIVE RH-11 INITIALIZE
4133                                                            ;SETUP UNIT NUMBER
4134
4135                                    ;*THIS CODE CHECKS TO SEE IF MANUAL INTERVENTION TESTS ARE OK
4136
4137  016266  005737  004724            TST     @#NOPUSH        ;IS THIS A 220 START ?
4138  016272  001007                    BNE     1$              ;SKIP THIS TEST IF SO
4139  016274  005737  000042            TST     @#42            ;MONITOR (ACT 11) RETURN ADDRESS ?
4140  016300  001004                    BNE     1$              ;SKIP THIS TEST
4141  016302  005737  001100            TST     @#$PASS         ;FIRST PASS ?
4142  016306  001001                    BNE     1$              ;SKIP THIS TEST IF NOT
4143  016310  000402                    BR      2$              ;CONTINUE WITH THIS TEST
4144
4145  016312                    1$:
4146  016312  000137  017012            JMP     TST21   ;       JUMP TO NEXT TEST ----------------)
4147  016316                    2$:
4148  016316  104401  016324            TYPE    ,65$            ;;TYPE ASCIZ STRING
4149  016322  000407                    BR      64$             ;;GET OVER THE ASCIZ
4150                            ;;65$:  .ASCIZ  <15><12>/STOP DRIVE /
4151  016342                    64$:
4152
```

```
4153  016342  013746  004716                  MOV     @#UNIT,-(SP)    ;GET UNIT UNDER TEST
4154  016346  104405                          TYPDS                   ;TYPE IT OUT
4155  016350  104401  001223                  TYPE    ,$CRLF
4156  016354  032713  010000          3$:     BIT     #MOL,@R3        ;MOL WILL BE HIGH TILL STOP IS HIT
4157  016360  001375                          BNE     3$              ;WAIT TILL STOP IS HIT
4158
4159  016362  104401  016370                  TYPE    ,67$            ;;TYPE ASCIZ STRING
4160  016366  000406                          BR      66$             ;;GET OVER THE ASCIZ
4161                          ;;67$:  .ASCIZ  /START DRIVE/
4162  016404                  66$:
4163
4164  016404  013746  004716                  MOV     @#UNIT,-(SP)    ;GET UNIT UNDER TEST
4165  016410  104405                          TYPDS                   ;TYPE IT
4166  016412  104401  001223                  TYPE    ,$CRLF
4167
4168  016416  032713  010000          4$:     BIT     #MOL,@R3        ;MOL WILL BE LOW TILL FILE READY
4169  016422  001775                          BEQ     4$              ;WAIT TILL FILE READY
4170
4171  016424  004737  041524                  JSR     PC,@#CLDISK     ;SET R1-RHCS1, R2-RHCS2
4172                                                                  ;R3-RHDS1, R4-RHER1
4173                                                                  ;GIVE RH-11 INITIALIZE
4174                                                                  ;SETUP UNIT NUMBER
4175  016430  004737  041562                  JSR     PC,@#CHECK      ;CHECK THAT DVA,RDY,MOL,DPR,DRY = 1
4176                                                                  ;AND THAT NO STATUS BITS ARE STUCK = 1
4177  016434  104401  067033                  TYPE    ,CPHALT         ;CANNOT CONTINUE TESTS IF THEY AREN'T
4178  016440  000000                          HALT                    ;STOP
4179  016442  012700  002272                  MOV     #RHWC,R0        ;ADDR. OF ADDR. OF RHWC IN R0
4180
4181
4182                                          ;*NOW INITIALIZE ALL THE REGISTERS
4183
4184  016446  012730  177777                  MOV     #177777,@(R0)+  ;LOAD 177777 INTO RHWC
4185
4186  016452  012730  177777                  MOV     #177777,@(R0)+  ;LOAD 177777 INTO RHBA
4187
4188  016456  052730  043010                  BIS     #43010,@(R0)+   ;LOAD 43010 INTO RHCS2
4189  016462  012730  001400                  MOV     #1400,@(R0)+    ;LOAD 1400 INTO RHCS1
4190
4191  016466  012730  000000                  MOV     #0,@(R0)+       ;LOAD 0 INTO RHER1
4192
4193  016472  012730  177777                  MOV     #177777,@(R0)+  ;LOAD 177777 INTO RHDST
4194
4195  016476  012730  000000                  MOV     #0,@(R0)+       ;LOAD 0 INTO RHER2
4196
4197  016502  012730  177777                  MOV     #177777,@(R0)+  ;LOAD 177777 INTO RHOF
4198
4199  016506  012730  177777                  MOV     #177777,@(R0)+  ;LOAD 177777 INTO RHCA
4200
4201  016512  012730  000000                  MOV     #0,@(R0)+       ;LOAD 0 INTO RHER3
4202
4203                                          ;*NOW SET BITS IN RHAS FOR ALL DRIVES PRESENT
4204
4205  016516  010046                          MOV     R0,-(SP)        ;;PUSH R0 ON STACK
4206  016520  011446                          MOV     @R4,-(SP)       ;SAVE RHER1 TO REINSTATE LATER
4207  016522  011246                          MOV     @R2,-(SP)       ;SAVE RHCS2 TO BE REINSTATED
4208                                                                  ;AFTER ALL ATA BITS HAVE BEEN SET
```

```
 4209  016524  013700  004742                MOV     @#TOTALAT,R0    ;GET DRIVES PRESENT
 4210  016530  005012                        CLR     @R2             ;CLEAR RHCS2 AND CARRY
 4211  016532  012705  000010                MOV     #8.,R5          ;COUNTER
 4212  016536  006000                88$:    ROR     R0              ;GET BIT INTO CARRY
 4213  016540  103002                        BCC     89$             ;BRANCH IF NO UNIT ON THIS BIT
 4214  016542  012714  177777                MOV     #-1,@R4         ;MOVE INTO ERROR REGISTER
 4215                                                                ;TO SET ATA BIT
 4216  016546  005212                89$:    INC     @R2             ;INCREMENT RHCS2 TO NEXT UNIT
 4217  016550  005305                        DEC     R5              ;COUNT
 4218  016552  001371                        BNE     88$             ;BRANCH IF 8 NOT DONE
 4219  016554  012612                        MOV     (SP)+,@R2       ;REINSTATE RHCS2
 4220  016556  012614                        MOV     (SP)+,@R4       ;REINSTATE RHER1
 4221  016560  012600                        MOV     (SP)+,R0        ;;POP STACK INTO R0
 4222  016562  005720                        TST     (R0)+           ;GET OVER PHAS IN R0
 4223
 4224
 4225  016564  012730  177776                MOV     #177776,@(R0)+  ;LOAD 177776 INTO RHMR
 4226
 4227
 4228  016570  013777  004740  163520        MOV     @#ATTENT,@RHAS  ;CLEAR WORKING DRIVE 'ATA'
 4229  016576  013777  002462  163474        MOV     @#READIN,@RHCS1 ;GET READY FOR READIN
 4230                                                                ;READ IN WITH 20 IN RHCS1
 4231
 4232
 4233                                        ;*NOW SAVE REGISTERS FOR COMPARISON AFTER READ-IN COMMAND
 4234
 4235  016604  004037  041672                JSR     R0,@#SAVER      ;SAVE REGISTERS
 4236  016610  002272                        RHWC                    ;RHWC IS THE FIRST REGISTER SAVED
 4237  016612  004612                        SAVERE                  ;STARTING ADDRESS OF WHERE
 4238                                                                ;THE REGISTERS ARE SAVED
 4239  016614  000022                        18.                     ;NUMBER OF REGISTERS
 4240                                                                ;SAVED = 18.
 4241
 4242  016616  013777  004606  163442        MOV     @#RP4VEC,@RPVEC ;SET RP04 VECTOR ADDRESS
 4243                                                                ;TO 'TIME1' IF P-CLOCK IS PRESENT
 4244                                                                ;OR TO 'TIME2' IF P-CLOCK IS NOT PRESENT
 4245                                                                ;'TIME' WILL ONLY SAVE
 4246                                                                ;CURRENT CYLINDER ADDRESS
 4247                                                                ;AND LOOK AHEAD REGISTERS
 4248
 4249  016624  005737  004750                TST     @#RH70          ;RUNNING ON AN RH70 CONTROLLER ?
 4250  016630  001411                        BEQ     7$              ;SKIP NEXT FOR RH70 IF NOT
 4251
 4252  016632  013746  002462                MOV     @#READIN,-(SP)  ;GET READY TO MOVE COMMAND
 4253  016636  052716  000001                BIS     #GO,(SP)        ;GET READY TO SET GO
 4254                                                                ;WITHOUT INTERRUPT ENABLE
 4255  016642  012677  163432                MOV     (SP)+,@RHCS1    ;GO WITH
 4256                                                                 ;20 IN RHCS1 FOR READ IN
 4257                                                                ;WITH INTERRUPT DISABLED
 4258
 4259  016646  011100                        MOV     @R1,R0          ;SAVE RHCS1 DURING ABOVE OPERATION
 4260  016650  011305                        MOV     @R3,R5          ;SAVE RHDS1 DURING ABOVE OPERATION
 4261  016652  000406                        BR      8$              ;SKIP NEXT FOR RH11
 4262
 4263  016654                        7$:
 4264
```

```
4265  016654  013746  002462                  MOV     @#READIN,-(SP)  ;GET READY TO MOVE COMMAND
4266  016660  052716  000101                  BIS     #GO!IE,(SP)     ;GET READY TO SET 'GO' AND
4267                                                                    ;ENABLE INTERRUPT
4268  016664  012677  163410                  MOV     (SP)+,@RHCS1    ;GO WITH
4269                                                                     ;20 IN RHCS1 FOR READ IN
4270                                                                    ;WITH INTERRUPT ENABLED
4271
4272  016670                          8$:
4273
4274  016670  104413                          WAT                     ;WAIT FOR VV BIT TO SET
4275  016672  002322                          RHDS1                   ;WAIT FOR RHDS1 REGISTER
4276  016674  000100                          VV                      ;WAIT FOR VV BIT IN RHDS1 REGISTER
4277  016676  000001                          1.                      ;ALLOW 10 MICRO SECONDS
4278  016700  000001                          1.                      ;VV MUST SET BETWEEN
4279                                                                    ;00 AND 20 MICRO SECONDS
4280
4281                                          ;*NOW CHANGE SAVED REGISTERS TO EXPECTED VALUE
4282
4283  016702  004037  041426                  JSR     R0,@#FILLRE     ;MOV 0 INTO SAVED RHCA
4284  016706  002312                          RHCA                    ;SAVED REGISTER TO CHANGE
4285  016710  000000                          0                       ;DATA
4286  016712  004037  041426                  JSR     R0,@#FILLRE     ;MOV 0 INTO SAVED RHDST
4287  016716  002304                          RHDST                   ;SAVED REGISTER TO CHANGE
4288  016720  000000                          0                       ;DATA
4289
4290  016722  004037  042404                  JSR     R0,@#CHREG      ;CHANGE BITS IN SAVED REGISTER
4291  016726  002310                          RHOF                    ;CHANGE RHOF REGISTER
4292
4293  016730  000003                          3                       ;3 BIT/BITS TO BE CHANGED
4294  016732  000000                          0                       ;NEW VALUE OF FMT22 IS 0
4295  016734  010000                          FMT22                   ;CHANGE FMT22 BIT
4296  016736  000000                          0                       ;NEW VALUE OF ECI IS 0
4297  016740  004000                          ECI                     ;CHANGE ECI BIT
4298  016742  000000                          0                       ;NEW VALUE OF HCI IS 0
4299  016744  002000                          HCI                     ;CHANGE HCI BIT
4300
4301  016746  004037  042404                  JSR     R0,@#CHREG      ;CHANGE BITS IN SAVED REGISTER
4302  016752  002322                          RHDS1                   ;CHANGE RHDS1 REGISTER
4303
4304  016754  000001                          1                       ;1 BIT/BITS TO BE CHANGED
4305  016756  000001                          1                       ;NEW VALUE OF VV IS 1
4306  016760  000100                          VV                      ;CHANGE VV BIT
4307
4308                                          ;*NOW THAT SAVERE TABLE WITH SAVED REGISTERS HAVE
4309                                          ;*THE EXPECTED VALUE AFTER A READ-IN COMMAND
4310                                          ;*COMPARISONS ARE MADE
4311
4312
4313  016762  004037  042512                  JSR     R0,@#COMREG     ;COMPARE SAVED REGISTERS WITH
4314                                                                    ;PRESENT VALUE
4315  016766  004612                          SAVERE                  ;GOOD DATA SAVED IN 'SAVERE'
4316  016770  002354                          WC                      ;TEST DATA STARTING FROM 'RHWC'
4317  016772  000022                          18.                     ;18. REGISTERS TO BE COMPARED
4318  016774  017000                          5$                      ;RETURN TO 5$ ON ERROR
4319  016776  017004                          6$                      ;RETURN TO 6$ ON NO ERROR
4320
```

```
 4321  017000  104020                  5$:     ERROR   20              ;READ IN COMMAND GAVE AN
 4322  017002  000207                          RTS     PC              ;ERROR
 4323                                                                  ;GOOD DATA HAS WHAT SHOULD
 4324                                                                  ;BE IN REGISTER AFTER A
 4325                                                                  ;READ-IN COMMAND
 4326                                                                  ;RECEIVED DATA HAS WHAT
 4327                                                                  ;THE REGISTER ACTUALLY CONTAINED
 4328                                                                  ;THE FOLLOWING SHOULD
 4329                                                                  ;BE THE REGISTER CONTENTS
 4330                                                                  ;RHCA=0, RHDST = 0
 4331                                                                  ;RHOF SHOULD HAVE FMT22 = 0,
 4332                                                                  ;HCI = 0, ECI = 0,
 4333                                                                  ;RHDS1 SHOULD HAVE VV = 1
 4334                                                                  ;ALL OTHER BITS SHOULD
 4335                                                                  ;BE UNCHANGED
 4336  017004  012737  177777  047342  6$:     MOV     #-1,@#PRITEM    ;CLEAR PREVIOUS ITEM NUMBER
```

```
                                ;;*****************************************************************
                                ;*TEST 21       READ-IN-PRESET
                                ;*
                                ;*      THIS TEST IS THE SAME AS THE PREVIOUS TEST EXCEPT
                                ;*      THAT IT DOES NOT TEST THE SETTING OF VV
                                ;*
                                ;*      THIS TEST IS HERE BECAUSE IF ACT-11 MONITOR IS PRESENT
                                ;*      THEN THE PREVIOUS TEST WILL NOT BE PERFORMED
                                ;*      THIS TESTS THE READ-IN-PRESET COMMAND
                                ;*      ALL WRITE BITS OF ALL REGISTERS ARE FILLED
                                ;*      WITH ONES EXCEPT GO,CLR,IE,PAT,MCPE,UPE
                                ;*      ATA FOR DRIVE UNDER TEST IS MADE = 0
                                ;*
                                ;*      THEN A READ-IN-PRESET COMMAND = 20 IS GIVEN
                                ;*      THEN ALL REGISTERS ARE TESTED
                                ;*      THE FOLLOWING SHOULD BE THE REGISTER CONTENTS
                                ;*      RHCA = 0, RHDST = 0, RHOF SHOULD HAVE
                                ;*      FMT22 = 0, ECI = 0, HCI = 0, RHDS1 SHOULD HAVE VV = 1
                                ;*      ALL OTHER REGISTERS SHOULD BE UNCHANGED
                                ;;*****************************************************************
017012  000004                  TST21:  SCOPE
017014  012706  001000                  MOV     #STACK,SP       ;RESET STACK
017020  012737  000021  004604          MOV     #21,@#TSTNM     ;SAVE TEST NUMBER
017026  004737  041524                  JSR     PC,@#CLDISK     ;SET R1-RHCS1, R2-RHCS2
                                                                ;R3-RHDS1, R4-RHER1
                                                                ;GIVE RH-11 INITIALIZE
                                                                ;SETUP UNIT NUMBER
017032  004737  041562                  JSR     PC,@#CHECK      ;CHECK THAT DVA,RDY,MOL,DPR,DRY = 1
                                                                ;AND THAT NO STATUS BITS ARE STUCK = 1
017036  104401  067033                  TYPE    ,CPHALT         ;CANNOT CONTINUE TESTS IF THEY AREN'T
017042  000000                          HALT                    ;STOP
017044  012700  002272                  MOV     #RHWC,R0        ;ADDR. OF ADDR. OF RHWC IN R0

                                ;*INITIALIZE ALL THE REGISTERS

017050  012730  177777                  MOV     #177777,@(R0)+  ;LOAD 177777 INTO RHWC

017054  012730  177777                  MOV     #177777,@(R0)+  ;LOAD 177777 INTO RHBA

017060  052730  043010                  BIS     #43010,@(R0)+   ;LOAD 43010 INTO RHCS2
017064  012730  001400                  MOV     #1400,@(R0)+    ;LOAD 1400 INTO RHCS1

017070  012730  000000                  MOV     #0,@(R0)+       ;LOAD 0 INTO RHER1

017074  012730  177777                  MOV     #177777,@(R0)+  ;LOAD 177777 INTO RHDST

017100  012730  000000                  MOV     #0,@(R0)+       ;LOAD 0 INTO RHER2

017104  012730  177777                  MOV     #177777,@(R0)+  ;LOAD 177777 INTO RHOF

017110  012730  177777                  MOV     #177777,@(R0)+  ;LOAD 177777 INTO RHCA
```

```
                                        MOV     #0,@(R0)+       ;LOAD 0 INTO RHER3
                                        MOV     #-1,@(R0)+      ;CLEAR ALL BITS OF RHAS
                                        MOV     #177776,@(R0)+  ;LOAD 177776 INTO RHMR

                                        MOV     @#ATTENT,@RHAS  ;CLEAR WORKING DRIVE 'ATA'
                                        MOV     @#READIN,@RHCS1 ;GET READY FOR READIN
                                                                ;READ IN WITH 20 IN RHCS1


                                        ;*NOW SAVE REGISTERS FOR COMPARISON AFTER READ-IN COMMAND

                                        JSR     R0,@#SAVER      ;SAVE REGISTERS
                                        RHWC                    ;RHWC IS THE FIRST REGISTER SAVED
                                        SAVERE                  ;STARTING ADDRESS OF WHERE
                                                                ;THE REGISTERS ARE SAVED
                                        18.                     ;NUMBER OF REGISTERS
                                                                ;SAVED = 18.

                                        MOV     @#RP4VEC,@RPVEC ;SET RP04 VECTOR ADDRESS
                                                                ;TO 'TIME1' IF P-CLOCK IS PRESENT
                                                                ;OR TO 'TIME2' IF P-CLOCK IS NOT PRESENT
                                                                ;'TIME' WILL ONLY SAVE
                                                                ;CURRENT CYLINDER ADDRESS
                                                                ;AND LOOK AHEAD REGISTERS

                                        TST     @#RH70          ;RUNNING ON AN RH70 CONTROLLER ?
                                        BEQ     9$              ;SKIP NEXT IF NOT

                                        MOV     @#READIN,-(SP)  ;GET READY TO MOVE COMMAND
                                        BIS     #GO,(SP)        ;GET READY TO SET GO
                                                                ;WITHOUT INTERRUPT ENABLE
                                        MOV     (SP)+,@RHCS1    ;GO WITH
                                                                 ;20 IN RHCS1 FOR READ IN
                                                                ;WITH INTERRUPT DISABLED

                                        MOV     @R1,R0          ;SAVE RHCS1 DURING ABOVE OPERATION
                                        MOV     @R3,R5          ;SAVE RHDS1 DURING ABOVE OPERATION
                                        BR      10$             ;SKIP NEXT RH11 CODE

                                9$:

                                        MOV     @#READIN,-(SP)  ;GET READY TO MOVE COMMAND
                                        BIS     #GO!IE,(SP)     ;GET READY TO SET 'GO' AND
                                                                ;ENABLE INTERRUPT
                                        MOV     (SP)+,@RHCS1    ;GO WITH
                                                                 ;20 IN RHCS1 FOR READ IN
                                                                ;WITH INTERRUPT ENABLED


                                10$:

                                        WAT                     ;WAIT FOR RDY BIT TO SET
                                        RHCS1                   ;WAIT FOR RHCS1 REGISTER
                                        RDY                     ;WAIT FOR RDY BIT IN RHCS1 REGISTER
```

```
 4449  017236  000001                    1.                      ;ALLOW 10 MICRO SECONDS
 4450  017240  000001                    1.                      ;RDY MUST SET BETWEEN
 4451                                                            ;00 AND 20 MICRO SECONDS
 4452
 4453                                    ;*NOW CHANGE SAVED REGISTERS TO EXPECTED VALUE
 4454
 4455  017242  004037  041426           JSR     R0,@#FILLRE     ;MOV 0 INTO SAVED RHCA
 4456  017246  002312                    RHCA                    ;SAVED REGISTER TO CHANGE
 4457  017250  000000                    0                       ;DATA
 4458  017252  004037  041426           JSR     R0,@#FILLRE     ;MOV 0 INTO SAVED RHDST
 4459  017256  002304                    RHDST                   ;SAVED REGISTER TO CHANGE
 4460  017260  000000                    0                       ;DATA
 4461
 4462  017262  004037  042404           JSR     R0,@#CHREG      ;CHANGE BITS IN SAVED REGISTER
 4463  017266  002310                    RHOF                    ;CHANGE RHOF REGISTER
 4464
 4465  017270  000003                    3                       ;3 BIT/BITS TO BE CHANGED
 4466  017272  000000                    0                       ;NEW VALUE OF FMT22 IS 0
 4467  017274  010000                    FMT22                   ;CHANGE FMT22 BIT
 4468  017276  000000                    0                       ;NEW VALUE OF ECI IS 0
 4469  017300  004000                    ECI                     ;CHANGE ECI BIT
 4470  017302  000000                    0                       ;NEW VALUE OF HCI IS 0
 4471  017304  002000                    HCI                     ;CHANGE HCI BIT
 4472
 4473  017306  004037  042404           JSR     R0,@#CHREG      ;CHANGE BITS IN SAVED REGISTER
 4474  017312  002322                    RHDS1                   ;CHANGE RHDS1 REGISTER
 4475
 4476  017314  000001                    1                       ;1 BIT/BITS TO BE CHANGED
 4477  017316  000001                    1                       ;NEW VALUE OF VV IS 1
 4478  017320  000100                    VV                      ;CHANGE VV BIT
 4479
 4480                                    ;*NOW THAT SAVERE TABLE WITH SAVED REGISTERS HAVE
 4481                                    ;*THE EXPECTED VALUE AFTER A READ-IN COMMAND
 4482                                    ;*COMPARISONS ARE MADE
 4483
 4484
 4485  017322  004037  042512           JSR     R0,@#COMREG     ;COMPARE SAVED REGISTERS WITH
 4486                                                            ;PRESENT VALUE
 4487  017326  004612                    SAVERE                  ;GOOD DATA SAVED IN 'SAVERE'
 4488  017330  002354                    WC                      ;TEST DATA STARTING FROM 'RHWC'
 4489  017332  000022                    18.                     ;18. REGISTERS TO BE COMPARED
 4490  017334  017340                    5$                      ;RETURN TO 5$ ON ERROR
 4491  017336  017344                    6$                      ;RETURN TO 6$ ON NO ERROR
 4492
 4493  017340  104020            5$:     ERROR   20              ;READ-IN COMMAND GAVE AN
 4494  017342  000207                    RTS     PC              ;ERROR
 4495                                                            ;GOOD DATA HAS WHAT SHOULD
 4496                                                            ;BE IN REGISTER AFTER A
 4497                                                            ;READ-IN COMMAND
 4498                                                            ;RECEIVED DATA HAS WHAT
 4499                                                            ;THE REGISTER ACTUALLY CONTAINE
 4500                                                            ;THE FOLLOWING SHOULD
 4501                                                            ;BE THE REGISTER CONTENTS
 4502                                                            ;RHCA = 0, RHDST = 0
 4503                                                            ;RHOF SHOULD HAVE FMT22 = 0,
 4504                                                            ;HCI = 0, ECI = 0,
```

```
 4505                                                ;RHDS1 SHOULD HAVE VV = 1
 4506                                                ;ALL OTHER BITS SHOULD
 4507                                                ;BE UNCHANGED
 4508  017344                        6$:
```

```
                                ;;*****************************************************************
017344  005737  004744          TST     @#RP06          ;TEST FOR RP06 DRIVE
017350  001401                  BEQ     7$              ;IF = 0, TREAT DRIVE AS AN RP04
017352  000402                  BR      8$              ;TREAT AS RP06 - DO NEXT "MAKECL"
017354  000137  017420          7$:     JMP     @#DOG           ;DO SECOND FOLLOWING "MAKECL"
017360                          8$:
                                ;;*****************************************************************
                                ;;*****************************************************************
                                ;*TEST 22        MAKE CURRENT CYLINDER = 777
                                ;;*****************************************************************
017360  000004                  TST22:  SCOPE
017362  012706  001000                  MOV     #STACK,SP       ;RESET STACK
017366  012737  000022  004604          MOV     #23-1,@#TSTNM   ;THIS SAVES TEST NUMBER

017374  004737  041524          JSR     PC,@#CLDISK     ;INIT DRIVE
017400  012777  000001  162712  MOV     #DMD,@RHMR      ;SET DIAGNOSTIC MODE
017406  004037  041174          JSR     R0,@#MAKECYL    ;SUBROUTINE TO GIVE A SEEK
                                                        ;COMMAND FOLLOWED BY A INIT
                                                        ;THIS SHOULD CHANGE RHCC TO 777
017412  000777                  777

017414  000137  017454          JMP     @#FISH          ;DON'T DO NEXT "MAKECL"

017420                          DOG:

                                ;;*****************************************************************
                                ;*TEST 23        MAKE CURRENT CYLINDER = 377
                                ;;*****************************************************************
017420  000004                  TST23:  SCOPE
017422  012706  001000                  MOV     #STACK,SP       ;RESET STACK
017426  012737  000023  004604          MOV     #24-1,@#TSTNM   ;THIS SAVES TEST NUMBER

017434  004737  041524          JSR     PC,@#CLDISK     ;INIT DRIVE
017440  012777  000001  162652  MOV     #DMD,@RHMR      ;SET DIAGNOSTIC MODE
017446  004037  041174          JSR     R0,@#MAKECYL    ;SUBROUTINE TO GIVE A SEEK
                                                        ;COMMAND FOLLOWED BY A INIT
                                                        ;THIS SHOULD CHANGE RHCC TO 377
017452  000377                  377
```

```
                                ;;*****************************************************************
                                ;*TEST 24       RECALIBRATE COMMAND

                                ;*      ALL POSSIBLE REGISTERS ARE FILLED WITH ONES
                                ;*      THEN A RECALIBRATE = 6 COMMAND IS GIVEN

                                ;*      NO REGISTERS SHOULD CHANGE EXCEPT RHCC = 0

                                ;;*****************************************************************
017454  000004                  TST24:  SCOPE
017456  012706  001000                  MOV     #STACK,SP       ;RESET STACK
017462  012737  000024  004604          MOV     #24,@#TSTNM     ;SAVE TEST NUMBER

017470  004737  041524                  JSR     PC,@#CLDISK     ;SET R1-RHCS1, R2-RHCS2
                                                                ;R3-RHDS1, R4-RHER1
                                                                ;GIVE RH-11 INITIALIZE
                                                                ;SETUP UNIT NUMBER
017474  004737  041604                  JSR     PC,@#CHECKT     ;CHECK DVA,RDY,MOL,DPR,DRY,VV = 1
                                                                ;AND THAT NO STATUS BITS ARE STUCK = 1
017500  104401  067033                  TYPE    ,CPHALT         ;CANNOT CONTINUE TESTING IF ANY OF
                                                                ;THE FIRST SET OF BITS DON'T = 1
017504  000000                          HALT                    ;STOP
017506  012700  002272                  MOV     #RHWC,R0        ;ADDR. OF ADDR. OF RHWC IN R0
017512  012730  177777                  MOV     #177777,@(R0)+  ;LOAD 177777 INTO RHWC

017516  012730  177776                  MOV     #177776,@(R0)+  ;LOAD 177776 INTO RHBA

017522  052730  000010                  BIS     #010,@(R0)+     ;LOAD 010 INTO RHCS2
017526  012730  001400                  MOV     #1400,@(R0)+    ;LOAD 1400 INTO RHCS1

017532  012730  000000                  MOV     #0,@(R0)+       ;LOAD 0 INTO RHER1

017536  012730  177777                  MOV     #177777,@(R0)+  ;LOAD 177777 INTO RHDST

017542  012730  000000                  MOV     #0,@(R0)+       ;LOAD 0 INTO RHER2

017546  012730  177777                  MOV     #177777,@(R0)+  ;LOAD 177777 INTO RHOF

017552  012730  177777                  MOV     #177777,@(R0)+  ;LOAD 177777 INTO RHCA

017556  012730  000000                  MOV     #0,@(R0)+       ;LOAD 0 INTO RHER3

                                        ;*NOW SET BITS IN RHAS FOR ALL DRIVES PRESENT

017562  010046                          MOV     R0,-(SP)        ;;PUSH R0 ON STACK
017564  011446                          MOV     @R4,-(SP)       ;SAVE RHER1 TO REINSTATE LATER
017566  011246                          MOV     @R2,-(SP)       ;SAVE RHCS2 TO BE REINSTATED
                                                                ;AFTER ALL ATA BITS HAVE BEEN SET
017570  013700  004742                  MOV     @#TOTALAT,R0    ;GET DRIVES PRESENT
017574  005012                          CLR     @R2             ;CLEAR RHCS2 AND CARRY
017576  012705  000010                  MOV     #8.,R5          ;COUNTER
017602  006000          84$:            ROR     R0              ;GET BIT INTO CARRY
```

```
4610  017604  103002                  BCC     85$             ;BRANCH IF NO UNIT ON THIS BIT
4611  017606  012714  177777          MOV     #-1,@R4         ;MOVE INTO ERROR REGISTER
4612                                                          ;TO SET ATA BIT
4613  017612  005212          85$:    INC     @R2             ;INCREMENT RHCS2 TO NEXT UNIT
4614  017614  005305                  DEC     R5              ;COUNT
4615  017616  001371                  BNE     84$             ;BRANCH IF 8 NOT DONE
4616  017620  012612                  MOV     (SP)+,@R2       ;REINSTATE RHCS2
4617  017622  012614                  MOV     (SP)+,@R4       ;REINSTATE RHER1
4618  017624  012600                  MOV     (SP)+,R0        ;;POP STACK INTO R0
4619  017626  005720                  TST     (R0)+           ;GET OVER PHAS IN R0
4620
4621
4622  017630  012730  177776          MOV     #177776,@(R0)+  ;LOAD 177776 INTO RHMR
4623
4624
4625  017634  013777  002426  162436  MOV     @#RECALI,@RHCS1 ;GET READY FOR RECALI
4626                                                          ;RECALIBRATE WITH 6 IN RHCS1
4627
4628
4629                                  ;*NOW SAVE REGISTERS FOR COMPARISON AFTER RECALIBRATE
4630
4631  017642  004037  041672          JSR     R0,@#SAVER      ;SAVE REGISTERS
4632  017646  002272                  RHWC                    ;RHWC IS THE FIRST REGISTER SAVED
4633  017650  004612                  SAVERE                  ;STARTING ADDRESS OF WHERE
4634                                                          ;THE REGISTERS ARE SAVED
4635  017652  000022                  18.                     ;NUMBER OF REGISTERS
4636                                                          ;SAVED = 18.
4637  017654  013777  004606  162404  MOV     @#RP4VEC,@RPVEC ;SET RP04 VECTOR ADDRESS
4638                                                          ;TO 'TIME1' IF P-CLOCK IS PRESENT
4639                                                          ;OR TO 'TIME2' IF P-CLOCK IS NOT PRESENT
4640                                                          ;'TIME' WILL ONLY SAVE
4641                                                          ;CURRENT CYLINDER ADDRESS
4642                                                          ;AND LOOK AHEAD REGISTERS
4643
4644  017662  013746  002426          MOV     @#RECALI,-(SP)  ;GET READY TO MOVE COMMAND
4645  017666  052716  000101          BIS     #GO!IE,(SP)     ;GET READY TO SET 'GO' AND
4646                                                          ;ENABLE INTERRUPT
4647  017672  012677  162402          MOV     (SP)+,@RHCS1    ;GO WITH
4648                                                          ;6 IN RHCS1 FOR RECALIBRATE
4649                                                          ;WITH INTERRUPT ENABLED
4650  017676  011100                  MOV     @R1,R0          ;SAVE RHCS1 DURING ABOVE OPERATION
4651  017700  011305                  MOV     @R3,R5          ;SAVE RHDS1 DURING ABOVE OPERATION
4652
4653  017702  104413                  WAT                     ;WAIT FOR DRY BIT TO SET
4654  017704  002322                  RHDS1                   ;WAIT FOR RHDS1 REGISTER
4655  017706  000200                  DRY                     ;WAIT FOR DRY BIT IN RHDS1 REGISTER
4656  017710  076377                  31999.                  ;ALLOW 319990 MICRO SECONDS
4657  017712  056701                  24001.                  ;DRY MUST SET BETWEEN
4658                                                          ;79980 AND 560000 MICRO SECONDS
4659
4660                                  ;*COMPARE CONTENTS OF RHCS1 AND RHDS1 ALREADY SAVED IN
4661                                  ;*R0 AND R5 IMMEDIATELY AFTER GO
4662
4663  017714  013746  002426          MOV     @#RECALI,-(SP)  ;SAVE COMMAND
4664  017720  052716  004301          BIS     #DVA!GO!IE!RDY,(SP)     ;INCLUDE DVA!GO!IE!RDY
4665  017724  005737  004722          TST     @#NUNIT         ;ARE THERE MORE THAN ONE UNIT
```

```
4666  017730  001413                          BEQ     89$             ;BRANCH IF ONLY ONE UNIT
4667  017732  010037  004760                  MOV     R0,@#TMP4       ;GET RHCS1
4668  017736  042737  177677  004760          BIC     #^C1E,@#TMP4    ;KEEP IE BIT
4669  017744  042716  000100                  BIC     #IE,(SP)        ;CLEAR IE IN GOOD DATA
4670  017750  053716  004760                  BIS     @#TMP4,(SP)     ;GET IE AS IS
4671  017754  052716  100000                  BIS     #SC,(SP)        ;SET SC IN RHCS1
4672  017760                          89$:
4673  017760  011637  001124                  MOV     (SP),@#$GDDAT   ;SAVE FOR PRINTOUT
4674  017764  022600                          CMP     (SP)+,R0        ;DURING ABOVE OPERATION ONLY DVA!GO!IE!RDY
4675                                                                  ;AND COMMAND SHOULD BE SET
4676  017766  001405                          BEQ     88$             ;BRANCH IF GOOD
4677  017770  010037  001126                  MOV     R0,@#$BDDAT     ;BAD DATA
4678  017774  010137  004600                  MOV     R1,@#REGADR     ;FAILING REGISTER RHCS1
4679  020000  104021                          ERROR   21              ;DURING ABOVE OPERATION ONLY
4680                                                                  ;COMMAND AND DVA!GO!IE!RDY SHOULD BE SET
4681  020002  012746  030500          88$:    MOV     #MOL!DPR!VV!PIP,-(SP)   ;SAVE BITS SET DURING OPERATION IN RHDS1
4682  020006  011637  001124                  MOV     (SP),@#$GDDAT   ;SAVE FOR PRINTOUT
4683  020012  022605                          CMP     (SP)+,R5        ;DURING ABOVE OPERATION ONLY MOL!DPR!VV!PIP
4684                                                                  ;SHOULD BE SET
4685  020014  001405                          BEQ     90$             ;BRANCH IF GOOD
4686  020016  010537  001126                  MOV     R5,@#$BDDAT     ;BAD DATA
4687  020022  010337  004600                  MOV     R3,@#REGADR     ;FAILING REGISTER RHDS1
4688  020026  104063                          ERROR   63              ;DURING ABOVE OPERATION ONLY
4689                                                                  ;MOL!DPR!VV!PIP SHOULD BE SET
4690  020030                          90$:
4691
4692                                  ;*NOW CHANGE SAVED REGISTERS TO EXPECTED VALUES
4693
4694  020030  004037  041426                  JSR     R0,@#FILLRE     ;MOV 0 INTO SAVED RHCC
4695  020034  002334                          RHCC                    ;SAVED REGISTER TO CHANGE
4696  020036  000000                          0                       ;DATA
4697  020040  004037  041426                  JSR     R0,@#FILLRE     ;MOV 116377 INTO SAVED RHOF
4698  020044  002310                          RHOF                    ;SAVED REGISTER TO CHANGE
4699  020046  116377                          116377                  ;DATA
4700
4701  020050  053737  004740  004636          BIS     @#ATTENT,@#SAVERE+24    ;SET APPROPRIATE 'ATA' BITS
4702                                                                  ;FOR WORKING DRIVE IN
4703                                                                  ;SAVED RHAS LOACTION
4704  020056  004037  041426                  JSR     R0,@#FILLRE     ;MOV 104206 INTO SAVED RHCS1
4705  020062  002300                          RHCS1                   ;SAVED REGISTER TO CHANGE
4706  020064  104206                          104206                  ;DATA
4707  020066  004037  041426                  JSR     R0,@#FILLRE     ;MOV 110700 INTO SAVED RHDS1
4708  020072  002322                          RHDS1                   ;SAVED REGISTER TO CHANGE
4709  020074  110700                          110700                  ;DATA
4710
4711
4712                                  ;*NOW COMPARE REGISTERS AFTER A RECALIBRATE COMMAND
4713
4714
4715  020076  004037  042512                  JSR     R0,@#COMREG     ;COMPARE SAVED REGISTERS WITH
4716                                                                  ;PRESENT VALUE
4717  020102  004012                          SAVERE                  ;GOOD DATA SAVED IN 'SAVERE'
4718  020104  002354                          WC                      ;TEST DATA STARTING FROM 'RHWC'
4719  020106  000022                          18.                     ;18. REGISTERS TO BE COMPARED
4720  020110  020114                          1$                      ;RETURN TO 1$ ON ERROR
4721  020112  020120                          2$                      ;RETURN TO 2$ ON NO ERROR
```

```
4722
4723  020114  104064          1$:   ERROR   64          ;RECALIBRATE COMMAND CAUSED
4724  020116  000207                RTS     PC          ;AN ERROR
4725                                                    ;GOOD DATA GIVES WHAT SHOULD
4726                                                    ;BE THERE
4727                                                    ;RECEIVED DATA GIVES WHAT WAS
4728                                                    ;THERE AFTER COMMAND
4729
4730  020120                    2$:
```

```
                                ;;*****************************************************************
020120  005737  004744          TST     @#RP06          ;TEST FOR RP06 DRIVE
020124  001401                  BEQ     3$              ;IF = 0, TREAT DRIVE AS AN RP04
020126  000402                  BR      4$              ;TREAT AS RP06 - DO NEXT "MAKECL"
020130  000137  020174      3$: JMP     @#CAT           ;DO SECOND FOLLOWING "MAKECL"
020134                      4$:
                                ;;*****************************************************************
                                ;;*****************************************************************
                                ;*TEST 25        MAKE CURRENT CYLINDER = 777
                                ;;*****************************************************************
020134  000004              TST25:  SCOPE
020136  012706  001000            MOV     #STACK,SP       ;RESET STACK
020142  012737  000025  004604    MOV     #26-1,@#TSTNM   ;THIS SAVES TEST NUMBER

020150  004737  041524            JSR     PC,@#CLDISK     ;INIT DRIVE
020154  012777  000001  162136    MOV     #DMD,@RHMR      ;SET DIAGNOSTIC MODE
020162  004037  041174            JSR     R0,@#MAKECYL    ;SUBROUTINE TO GIVE A SEEK
                                                        ;COMMAND FOLLOWED BY A INIT
                                                        ;THIS SHOULD CHANGE RHCC TO 777
020166  000777                    777

020170  000137  020230            JMP     @#BIRD          ;DON'T DO NEXT "MAKECL"


020174                      CAT:

                                ;;*****************************************************************
                                ;*TEST 26        MAKE CURRENT CYLINDER = 377
                                ;;*****************************************************************
020174  000004              TST26:  SCOPE
020176  012706  001000            MOV     #STACK,SP       ;RESET STACK
020202  012737  000026  004604    MOV     #27-1,@#TSTNM   ;THIS SAVES TEST NUMBER

020210  004737  041524            JSR     PC,@#CLDISK     ;INIT DRIVE
020214  012777  000001  162076    MOV     #DMD,@RHMR      ;SET DIAGNOSTIC MODE
020222  004037  041174            JSR     R0,@#MAKECYL    ;SUBROUTINE TO GIVE A SEEK
                                                        ;COMMAND FOLLOWED BY A INIT
                                                        ;THIS SHOULD CHANGE RHCC TO 377
020226  000377                    377

```

```
                                BIRD:
                                ;;*****************************************************************
                                ;*TEST 27       RECALIBRATE COMMAND

                                ;*      ALL POSSIBLE REGISTERS ARE FILLED WITH 0
                                ;*      THEN A RECALIBRATE =6 COMMAND IS GIVEN

                                ;*      NO REGISTERS SHOULD CHANGE EXCEPT RHCC=0

                                ;;*****************************************************************
020230  000004                  TST27:  SCOPE
020232  012706  001000                  MOV     #STACK,SP       ;RESET STACK
020236  012737  000027  004604          MOV     #27,@#TSTNM     ;SAVE TEST NUMBER

020244  004737  041524                  JSR     PC,@#CLDISK     ;SET R1-RHCS1, R2-RHCS2
                                                                ;R3-RHDS1, R4-RHER1
                                                                ;GIVE RH-11 INITIALIZE
                                                                ;SETUP UNIT NUMBER
020250  004737  041604                  JSR     PC,@#CHECKT     ;CHECK DVA,RDY,MOL,DPR,DRY,VV = 1
                                                                ;AND THAT NO STATUS BITS ARE STUCK = 1
020254  104401  067033                  TYPE    ,CPHALT         ;CANNOT CONTINUE TESTING IF ANY OF
                                                                ;THE FIRST SET OF BITS DON'T = 1
020260  000000                          HALT                    ;STOP
020262  012700  002272                  MOV     #RHWC,R0        ;ADDR. OF ADDR OF RHWC IN R0

020266  012730  000000                  MOV     #0,@(R0)+       ;LOAD 0 INTO RHWC

020272  012730  000000                  MOV     #0,@(R0)+       ;LOAD 0 INTO RHBA

020276  052730  000000                  BIS     #0,@(R0)+       ;LOAD 0 INTO RHCS2
020302  012730  000000                  MOV     #0,@(R0)+       ;LOAD 0 INTO RHCS1

020306  012730  000000                  MOV     #0,@(R0)+       ;LOAD 0 INTO RHER1

020312  012730  000000                  MOV     #0,@(R0)+       ;LOAD 0 INTO RHDST

020316  012730  000000                  MOV     #0,@(R0)+       ;LOAD 0 INTO RHER2

020322  012730  000000                  MOV     #0,@(R0)+       ;LOAD 0 INTO RHOF

020326  012730  000000                  MOV     #0,@(R0)+       ;LOAD 0 INTO RHCA

020332  012730  000000                  MOV     #0,@(R0)+       ;LOAD 0 INTO RHER3

020336  012730  177777                  MOV     #-1,@(R0)+      ;CLEAR ALL BITS OF RHAS
020342  012730  000000                  MOV     #0,@(R0)+       ;LOAD 0 INTO RHMR

020346  013777  002426  161724          MOV     @#RECALI,@RHCS1 ;GET READY FOR RECALI
                                                                ;RECALIBRATE WITH 6 IN RHCS1

                                ;*NOW SAVE REGISTERS FOR COMPARISON AFTER RECALIBRATE
```

```
 4831  020354  004037  041672                JSR     R0,@#SAVER      ;SAVE REGISTERS
 4832  020360  002272                        RHWC                    ;RHWC IS THE FIRST REGISTER SAVED
 4833  020362  004612                        SAVERE                  ;STARTING ADDRESS OF WHERE
 4834                                                                  ;THE REGISTERS ARE SAVED
 4835  020364  000022                        18.                     ;NUMBER OF REGISTERS
 4836                                                                  ;SAVED = 18.
 4837  020366  013777  004606  161672        MOV     @#RP4VEC,@RPVEC ;SET RP04 VECTOR ADDRESS
 4838                                                                  ;TO 'TIME1' IF P-CLOCK IS PRESENT
 4839                                                                  ;OR TO 'TIME2' IF P-CLOCK IS NOT PRESENT
 4840                                                                  ;'TIME' WILL ONLY SAVE
 4841                                                                  ;CURRENT CYLINDER ADDRESS
 4842                                                                  ;AND LOOK AHEAD REGISTERS
 4843
 4844  020374  013746  002426                MOV     @#RECALI,-(SP)  ;GET READY TO MOVE COMMAND
 4845  020400  052716  000101                BIS     #GO!IE,(SP)     ;GET READY TO SET 'GO' AND
 4846                                                                  ;ENABLE INTERRUPT
 4847  020404  012677  161670                MOV     (SP)+,@RHCS1    ;GO WITH
 4848                                                                  ;6 IN RHCS1 FOR RECALIBRATE
 4849                                                                  ;WITH INTERRUPT ENABLED
 4850  020410  011100                        MOV     @R1,R0          ;SAVE RHCS1 DURING ABOVE OPERATION
 4851  020412  011305                        MOV     @R3,R5          ;SAVE RHDS1 DURING ABOVE OPERATION
 4852
 4853  020414  104413                        WAT                     ;WAIT FOR DRY BIT TO SET
 4854  020416  002322                        RHDS1                   ;WAIT FOR RHDS1 REGISTER
 4855  020420  000200                        DRY                     ;WAIT FOR DRY BIT IN RHDS1 REGISTER
 4856  020422  076377                        31999.                  ;ALLOW 319990 MICRO SECONDS
 4857  020424  056701                        24001.                  ;DRY MUST SET BETWEEN
 4858                                                                  ;79980 AND 560000 MICRO SECONDS
 4859
 4860                                        ;*COMPARE CONTENTS OF RHCS1 AND RHDS1 ALREADY SAVED IN
 4861                                        ;*R0 AND R5 IMMEDIATELY AFTER GO
 4862
 4863  020426  013746  002426                MOV     @#RECALI,-(SP)  ;SAVE COMMAND
 4864  020432  052716  004301                BIS     #DVA!GO!IE!RDY,(SP)     ;INCLUDE DVA!GO!IE!RDY
 4865  020436  011637  001124                MOV     (SP),@#$GDDAT   ;SAVE FOR PRINTOUT
 4866  020442  022600                        CMP     (SP)+,R0        ;DURING ABOVE OPERATION ONLY DVA!GO!IE!RDY
 4867                                                                  ;AND COMMAND SHOULD BE SET
 4868  020444  001405                        BEQ     88$             ;BRANCH IF GOOD
 4869  020446  010037  001126                MOV     R0,@#$BDDAT     ;BAD DATA
 4870  020452  010137  004600                MOV     R1,@#REGADR     ;FAILING REGISTER RHCS1
 4871  020456  104021                        ERROR   21              ;DURING ABOVE OPERATION ONLY
 4872                                                                  ;COMMAND AND DVA!GO!IE!RDY SHOULD BE SET
 4873  020460  012746  030500        88$:    MOV     #MOL!DPR!VV!PIP,-(SP)   ;SAVE BITS SET DURING OPERATION IN RHDS1
 4874  020464  011637  001124                MOV     (SP),@#$GDDAT   ;SAVE FOR PRINTOUT
 4875  020470  022605                        CMP     (SP)+,R5        ;DURING ABOVE OPERATION ONLY MOL!DPR!VV!PIP
 4876                                                                  ;SHOULD BE SET
 4877  020472  001405                        BEQ     90$             ;BRANCH IF GOOD
 4878  020474  010537  001126                MOV     R5,@#$BDDAT     ;BAD DATA
 4879  020500  010337  004600                MOV     R3,@#REGADR     ;FAILING REGISTER RHDS1
 4880  020504  104063                        ERROR   63              ;DURING ABOVE OPERATION ONLY
 4881                                                                  ;MOL!DPR!VV!PIP SHOULD BE SET
 4882  020506                        90$:
 4883                                        ;*NOW CHANGE SAVED REGISTERS TO EXPECTED VALUES
 4884
 4885  020506  004037  041426                JSR     R0,@#FILLRE     ;MOV 0 INTO SAVED RHCC
 4886  020512  002334                        RHCC                    ;SAVED REGISTER TO CHANGE
```

```
4887  020514  000000                          0                       ;DATA
4888
4889  020516  053737  004740  004636          BIS     @#ATTENT,@#SAVERE+24    ;SET APPROPRIATE 'ATA' BITS
4890                                                                    ;FOR WORKING DRIVE IN
4891                                                                    ;SAVED RHAS LOACTION
4892
4893  020524  004037  042404                  JSR     R0,@#CHREG      ;CHANGE BITS IN SAVED REGISTER
4894  020530  002322                          RHDS1                   ;CHANGE RHDS1 REGISTER
4895
4896  020532  000001                          1                       ;1 BIT/BITS TO BE CHANGED
4897  020534  000001                          1                       ;NEW VALUE OF ATA IS 1
4898  020536  100000                          ATA                     ;CHANGE ATA BIT
4899
4900  020540  004037  042404                  JSR     R0,@#CHREG      ;CHANGE BITS IN SAVED REGISTER
4901  020544  002300                          RHCS1                   ;CHANGE RHCS1 REGISTER
4902
4903  020546  000001                          1                       ;1 BIT/BITS TO BE CHANGED
4904  020550  000001                          1                       ;NEW VALUE OF SC IS 1
4905  020552  100000                          SC                      ;CHANGE SC BIT
4906
4907                                          ;*NOW COMPARE REGISTERS AFTER A RECALIBRATE COMMAND
4908
4909
4910  020554  004037  042512                  JSR     R0,@#COMREG     ;COMPARE SAVED REGISTERS WITH
4911                                                                    ;PRESENT VALUE
4912  020560  004612                          SAVERE                  ;GOOD DATA SAVED IN 'SAVERE'
4913  020562  002354                          WC                      ;TEST DATA STARTING FROM 'RHWC'
4914  020564  000022                          18.                     ;18. REGISTERS TO BE COMPARED
4915  020566  020572                          1$                      ;RETURN TO 1$ ON ERROR
4916  020570  020576                          2$                      ;RETURN TO 2$ ON NO ERROR
4917
4918  020572  104064                  1$:     ERROR   64              ;RECALIBRATE COMMAND CAUSED
4919  020574  000207                          RTS     PC              ;AN ERROR
4920                                                                    ;GOOD DATA GIVES WHAT SHOULD BE
4921                                                                    ;THERE
4922                                                                    ;RECEIVED DATA GIVES WHAT WAS
4923  020576                          2$:                             ;THERE AFTER A RECALIBRATE
4924
```

```
                        ;;*****************************************************************
                        ;*TEST 30       UNLOAD COMMAND

                        ;*      IF STARTING ADDRESS 220 IS USED THIS TEST WILL NOT BE PERFORMED
                        ;*
                        ;*      IF THE PROGRAM WORKS UNDER ACT-11 MONITOR
                        ;*      THEN THIS TEST IS NOT PERFORMED
                        ;*
                        ;*      IF NO ACT-11 MONITOR IS PRESENT
                        ;*      THEN THIS TEST IS PERFORMED ONLY ON THE FIRST PASS
                        ;*      ON SUBSEQUENT PASSES THIS TEST IS NOT DONE
                        ;*
                        ;*
                        ;*      ALL POSSIBLE REGISTERS ARE FILLED WITH ONES
                        ;*      THEN AN UNLOAD COMMAND =2 IS GIVEN
                        ;*      NO REGISTERS SHOULD CHANGE EXCEPT MOL SHOULD=0
                        ;*      THEN THE DRIVE IS POWERED UP BY OPERATOR
                        ;*      AND A PACK ACKNOWLEDGE COMMAND (ALREADY TESTED)
                        ;*      SETS VV-IN RHDS1

                        ;;*****************************************************************
020576  000004          TST30:  SCOPE


                                ;*THIS CODE CHECKS TO SEE IF MANUAL INTERVENTION TESTS ARE OK

020600  005737  004724          TST     @#NOPUSH        ;IS THIS A 220 START ?
020604  001007                  BNE     1$              ;SKIP THIS TEST IF SO
020606  005737  000042          TST     @#42            ;MONITOR (ACT 11) RETURN ADDRESS ?
020612  001004                  BNE     1$              ;SKIP THIS TEST
020614  005737  001100          TST     @#$PASS         ;FIRST PASS ?
020620  001001                  BNE     1$              ;SKIP THIS TEST IF NOT
020622  000402                  BR      2$              ;CONTINUE WITH THIS TEST

020624                  1$:
020624  000137  022112          JMP     TST31   ;       JUMP TO NEXT TEST ----------------)
020630                  2$:
020630  012706  001000          MOV     #STACK,SP       ;RESET STACK
020634  012737  000030  004604  MOV     #30,@#TSTNM     ;SAVE TEST NUMBER

020642  004737  041524          JSR     PC,@#CLDISK     ;SET R1-RHCS1, R2-RHCS2
                                                        ;R3-RHDS1, R4-RHER1
                                                        ;GIVE RH-11 INITIALIZE
                                                        ;SETUP UNIT NUMBER
020646  004737  041604          JSR     PC,@#CHECKT     ;CHECK DVA,RDY,MOL,DPR,DRY,VV = 1
                                                        ;AND THAT NO STATUS BITS ARE STUCK = 1
020652  104401  067033          TYPE    ,CPHALT         ;CANNOT CONTINUE TESTING IF ANY OF
                                                        ;THE FIRST SET OF BITS DON'T = 1
020656  000000                  HALT                    ;STOP

                                ;*THIS SETTING OF VV IS FOR LOOP ON ERROR ONLY
                                ;*WHERE UNLOAD TAKES EFFECT AND CYCLE UP BRINGS VV DOWN

020660  017746  161436          MOV     @RHDS1,-(SP)    ;PUSH RHDS1 ONTO STACK
020664  042716  167677          BIC     #167677,(SP)    ;CLEAR EVERYTHING EXCEPT VV AND MOL
020670  022726  010100          CMP     #VV!MOL,(SP)+   ;ARE VV AND MOL SET ?
```

```
                                                BEQ     6$              ;CONTINUE IF YES
                                                TYPE    ,65$            ;;TYPE ASCIZ STRING
                                                BR      64$             ;;GET OVER THE ASCIZ
                                        ;;65$:  .ASCIZ  <15><12>/GET DRIVE HEADS LOADED THEN HIT "CONTINUE"/
                                        64$:
                                                TYPE    ,67$            ;;TYPE ASCIZ STRING
                                                BR      66$             ;;GET OVER THE ASCIZ
                                        ;;67$:  .ASCIZ  <15><12>/IF ALREADY LOADED THEN HIT "CONTINUE"/
                                        66$:
                                                HALT                    ;WAIT FOR CONTINUE


                                                JSR     PC,@#CLDISK     ;SET R1-RHCS1, R2-RHCS2
                                                                        ;R3-RHDS1, R4-RHER1
                                                                        ;GIVE RH-11 INITIALIZE
                                                                        ;SETUP UNIT NUMBER
                                                JSR     PC,@#CHECK      ;CHECK THAT DVA,RDY,MOL,DPR,DRY = 1
                                                                        ;AND THAT NO STATUS BITS ARE STUCK = 1
                                                TYPE    ,CPHALT         ;CANNOT CONTINUE TESTS IF THEY AREN'T
                                                HALT                    ;STOP

                                                ;*SET VV IN RHDS1 WITH PACK ACKNOWLEDGE


                                                MOV     @#PKACK,-(SP)   ;GET READY TO MOVE COMMAND
                                                BIS     #GO,(SP)        ;GET READY TO SET GO
                                                                        ;WITHOUT INTERRUPT ENABLE
                                                MOV     (SP)+,@RHCS1    ;GO WITH
                                                                         ;22 IN RHCS1 FOR PACK ACKNOWLEDGE
                                                                        ;WITH INTERRUPT DISABLED


                                                WAT                     ;WAIT FOR VV BIT TO SET
                                                RHDS1                   ;WAIT FOR RHDS1 REGISTER
                                                VV                      ;WAIT FOR VV BIT IN RHDS1 REGISTER
                                                1.                      ;ALLOW 10 MICRO SECONDS
                                                1.                      ;VV MUST SET BETWEEN
                                                                        ;00 AND 20 MICRO SECONDS

                                        6$:
                                                JSR     PC,@#CHECKT     ;CHECK DVA,RDY,MOL,DPR & VV = 1
                                                NOP                     ;CHECK THAT ALL OTHER BITS = 0
                                                NOP                     ;UNLIKE THE OTHER STATUS BIT TESTS,
                                                NOP                     ;THERE IS NO HALT IF IT FAILS - IT IS
                                                                        ;USED IN THE MIDDLE OF A SINGLE TEST

                                                MOV     #RHWC,R0        ;ADDR. OF ADDR OF RHWC IN R0

                                                ;*LOAD ALL POSSIBLE REGISTERS WITH ONES


                                                MOV     #177777,@(R0)+  ;LOAD 177777 INTO RHWC
```

```
5037  021130  012730  177777                MOV     #177777,@(R0)+  ;LOAD 177777 INTO RHBA
5038
5039
5040
5041  021134  052730  000010                BIS     #10,@(R0)+      ;LOAD 10 INTO RHCS2
5042
5043
5044  021140  012730  001400                MOV     #1400,@(R0)+    ;LOAD 1400 INTO RHCS1
5045
5046
5047
5048  021144  012730  000000                MOV     #0,@(R0)+       ;LOAD 0 INTO RHER1
5049
5050
5051
5052  021150  012730  177777                MOV     #177777,@(R0)+  ;LOAD 177777 INTO RHDST
5053
5054
5055
5056  021154  012730  000000                MOV     #0,@(R0)+       ;LOAD 0 INTO RHER2
5057
5058
5059
5060  021160  012730  177777                MOV     #177777,@(R0)+  ;LOAD 177777 INTO RH0F
5061
5062
5063
5064  021164  012730  177777                MOV     #177777,@(R0)+  ;LOAD 177777 INTO RHCA
5065
5066
5067
5068  021170  012730  000000                MOV     #0,@(R0)+       ;LOAD 0 INTO RHER3
5069
5070
5071
5072                                        ;*NOW SET BITS IN RHAS FOR ALL DRIVES PRESENT
5073
5074  021174  010046                        MOV     R0,-(SP)        ;;PUSH R0 ON STACK
5075  021176  011446                        MOV     @R4,-(SP)       ;SAVE RHER1 TO REINSTATE LATER
5076  021200  011246                        MOV     @R2,-(SP)       ;SAVE RHCS2 TO BE REINSTATED
5077                                                                ;AFTER ALL ATA BITS HAVE BEEN SET
5078  021202  013700  004742                MOV     @#TOTALAT,R0    ;GET DRIVES PRESENT
5079  021206  005012                        CLR     @R2             ;CLEAR RHCS2 AND CARRY
5080  021210  012705  000010                MOV     #8.,R5          ;COUNTER
5081  021214  006000              91$:      ROR     R0              ;GET BIT INTO CARRY
5082  021216  103002                        BCC     92$             ;BRANCH IF NO UNIT ON THIS BIT
5083  021220  012714  177777                MOV     #-1,@R4         ;MOVE INTO ERROR REGISTER
5084                                                                ;TO SET ATA BIT
5085  021224  005212              92$:      INC     @R2             ;INCREMENT RHCS2 TO NEXT UNIT
5086  021226  005305                        DEC     R5              ;COUNT
5087  021230  001371                        BNE     91$             ;BRANCH IF 8 NOT DONE
5088  021232  012612                        MOV     (SP)+,@R2       ;REINSTATE RHCS2
5089  021234  012614                        MOV     (SP)+,@R4       ;REINSTATE RHER1
5090  021236  012600                        MOV     (SP)+,R0        ;;POP STACK INTO R0
5091  021240  005720                        TST     (R0)+           ;GET OVER PHAS IN R0
5092
```

```
                                                ;*NOW SAVE REGISTERS FOR COMPARISON AFTER UNLOAD
```

```
021242  012730  177776          MOV     #177776,@(R0)+  ;LOAD 177776 INTO RHMR
```

```
021246  013777  002424  161024  MOV     @#UNLOAD,@RHCS1 ;GET READY FOR UNLOAD
                                                        ;UNLOAD WITH 2 IN RHCS1
```

```
021254  004037  041672          JSR     R0,@#SAVER      ;SAVE REGISTERS
021260  002272                  RHWC                    ;RHWC IS THE FIRST REGISTER SAVED
021262  004612                  SAVERE                  ;STARTING ADDRESS OF WHERE
                                                        ;THE REGISTERS ARE SAVED
021264  000021                  17.                     ;NUMBER OF REGISTERS
                                                        ;SAVED = 17.
```

```
021266  013777  004606  160772  MOV     @#RP4VEC,@RPVEC ;SET RP04 VECTOR ADDRESS
                                                        ;TO 'TIME1' IF P-CLOCK IS PRESENT
                                                        ;OR TO 'TIME2' IF P-CLOCK IS NOT PRESENT
                                                        ;'TIME' WILL ONLY SAVE
                                                        ;CURRENT CYLINDER ADDRESS
                                                        ;AND LOOK AHEAD REGISTERS
```

```
021274  013746  002424          MOV     @#UNLOAD,-(SP)  ;GET READY TO MOVE COMMAND
021300  052716  000001          BIS     #GO,(SP)        ;GET READY TO SET GO
                                                        ;WITHOUT INTERRUPT ENABLE
021304  012677  160770          MOV     (SP)+,@RHCS1    ;GO WITH
                                                        ;2 IN RHCS1 FOR UNLOAD
                                                        ;WITH INTERRUPT DISABLED
```

```
021310  011100                  MOV     @R1,R0          ;SAVE RHCS1 DURING ABOVE OPERATION
021312  011305                  MOV     @R3,R5          ;SAVE RHDS1 DURING ABOVE OPERATION
```

```
021314  104413                  WAT                     ;WAIT FOR BIT TO SET
021316  002300                  RHCS1                   ;IN RHCS1 REGISTER
021320  000200                  RDY                     ;'RDY' BIT
021322  000001                  1.                      ;ALLOW 10 MICRO SECONDS
021324  000001                  1.                      ;ANOTHER 10 - 'RDY' MUST SET BETWEEN
                                                        ;00 AND 20 MICRO SECONDS
```

```
5136
5137                                        ;*COMPARE CONTENTS OF RHCS1 AND RHDS1, WHICH WERE SAVED
5138                                        ;*DURING THE UNLOAD COMMAND, WITH THE EXPECTED RESULTS
5139
5140  021326  013746  002424                MOV     @#UNLOAD,-(SP)  ;PUSH COMMAND ON STACK
5141  021332  052716  004201                BIS     #DVA!GO!RDY,(SP) ;INCLUDE THESE BITS SET
5142  021336  005737  004722                TST     @#NUNIT         ;IS THERE MORE THAN ONE UNIT ?
5143  021342  001413                        BEQ     9$              ;SKIP NEXT IF ONLY ONE UNIT
5144  021344  010037  004760                MOV     R0,@#TMP4       ;PUT SAVED RHCS1 INTO TMP4
5145  021350  042737  177677  004760        BIC     #^C1E,@#TMP4    ;MASK ALL BUT THE 'IE' BIT IN RHCS1
5146  021356  042716  000100                BIC     #IE,(SP)        ;CLEAR 'IE' IN EXPECTED DATA
5147  021362  053716  004760                BIS     @#TMP4,(SP)     ;SET 'IE' STATE FROM ACTUAL RHCS1 DATA
5148  021366  052716  100000                BIS     #SC,(SP)        ;SET 'SC' IN RHCS1 SAVED DATA
5149
5150  021372  011637  001124        9$:     MOV     (SP),@#$GDDAT   ;SAVE EXPECTED DATA FOR PRINTOUT
5151  021376  022600                        CMP     (SP)+,R0        ;COMPARE EXPECTED DATA WITH SAVED
5152                                                                ;RHCS1 DATA AND RESET THE STACK
5153  021400  001405                        BEQ     10$             ;CHECK NEXT BITS IF THESE OK
5154  021402  010037  001126                MOV     R0,@#$BDDAT     ;RHCS1 IS BAD - PRINT IT OUT
5155  021406  010137  004600                MOV     R1,@#REGADR     ;REGISTER ADDRESS
5156  021412  104021                        ERROR   21              ;DURING ABOVE OPERATION ONLY THE
5157                                                                ;'DVA', 'GO', 'RDY' AND COMMAND BITS
5158                                                                ;SHOULD BE SET
5159
5160  021414  012746  020400        10$:    MOV     #PIP!DPR,-(SP)  ;PUT SOME EXPECTED RHDS1 BITS ON STACK
5161  021420  010537  004760                MOV     R5,@#TMP4       ;PUT SAVED RHDS1 INTO TMP4
5162  021424  042737  167677  004760        BIC     #^C<MOL!VV>,@#TMP4 ;MASK ALL BUT 'MOL' & 'VV' IN RHDS1
5163  021432  042716  010100                BIC     #MOL!VV,(SP)    ;CLEAR 'MOL' & 'VV' IN EXPECTED RHDS1
5164  021436  053716  004760                BIS     @#TMP4,(SP)     ;SET EXPECTED 'MOL' & 'VV' BIT STATES
5165                                                                ;FROM THE ACTUAL DATA (DON'T CARE)
5166  021442  011637  001124                MOV     (SP),@#$GDDAT   ;SAVE EXPECTED DATA FOR PRINTOUT
5167  021446  022605                        CMP     (SP)+,R5        ;COMPARE EXPECTED DATA WITH SAVED
5168                                                                ;RHDS1 DATA AND RESET THE STACK
5169  021450  001405                        BEQ     11$             ;CONTINUE IF EXPECTED=SAVED
5170  021452  010537  001126                MOV     R5,@#$BDDAT     ;RHDS1 IS BAD - PRINT IT OUT
5171  021456  010337  004600                MOV     R3,@#REGADR     ;REGISTER ADDRESS
5172  021462  104063                        ERROR   63              ;DURING THE ABOVE OPERATION, ONLY 'PIP
5173                                                                ;AND 'DPR' SHOULD BE SET
5174                                                                ;'MOL' & 'VV' ARE DON'T CARES
5175  021464                        11$:
5176
5177  021464  104401  021472                TYPE    ,96$            ;;TYPE ASCIZ STRING
5178  021470  000425                        BR      95$             ;;GET OVER THE ASCIZ
5179                                ;;96$:   .ASCIZ  <15><12>/IF STANDBY NOT LIT - ERROR AFTER UNLOAD/
5180  021544                        95$:
5181                                                                ;THIS PROVIDES A 1 SECOND "STALL"
5182
```

```
5183
5184
5185                                    ;*NOW CHANGE REGISTERS SAVED BEFORE UNLOAD COMMAND
5186                                    ;*TO EXPECTED VALUES AFTER UNLOAD COMMAND
5187
5188                                    ;* - AGAIN 'MOL' $ 'VV' ARE DON'T CARES
5189
5190
5191  021544  012746  020400            MOV     #PIP!DPR,-(SP)  ;SET EXPECTED FINAL RHDS1 BITS
5192  021550  017737  160546  004760    MOV     @RHDS1,@#TMP4   ;GET PRESENT ACTUAL RHDS1 CONTENTS
5193  021556  042737  167677  004760    BIC     #^C<MOL!VV>,@#TMP4 ;MASK OUT ALL BUT 'MOL' & 'VV'
5194  021564  042716  010100            BIC     #MOL!VV,(SP)    ;CLEAR 'MOL' & 'VV' IN EXPECTED RHDS1
5195  021570  053716  004760            BIS     @#TMP4,(SP)     ;SET EXPECTED 'MOL' & 'VV' STATES
5196                                            ;FROM THE ACTUAL (DON'T CARE COND.)
5197  021574  042716  100200            BIC     #ATA!DRY,(SP)   ;CLEAR THESE ADDITIONAL RHDS1 BITS
5198  021600  012637  004642            MOV     (SP)+,@#SAVERE+30 ;CHANGE THE SAVED RHDS1 REGISTER
5199                                            ;AND ADJUST THE STACK
5200
5201
5202  021604  004037  042404            JSR     R0,@#CHREG      ;CHANGE BITS IN SAVED REGISTER
5203  021610  002300                    RHCS1                   ;CHANGE RHCS1 REGISTER
5204
5205  021612  000001                    1                       ;1 BIT/BITS TO BE CHANGED
5206  021614  000001                    1                       ;NEW VALUE OF GO IS 1
5207  021616  000001                    GO                      ;CHANGE GO BIT
5208
5209  021620  005737  004722            TST     @#NUNIT         ;IS THERE MORE THAN ONE UNIT ?
5210  021624  001006                    BNE     7$              ;SKIP NEXT IF MORE THAN ONE UNIT
5211
5212  021626  004037  042404            JSR     R0,@#CHREG      ;CHANGE BITS IN SAVED REGISTER
5213  021632  002300                    RHCS1                   ;CHANGE RHCS1 REGISTER
5214
5215  021634  000001                    1                       ;1 BIT/BITS TO BE CHANGED
5216  021636  000000                    0                       ;NEW VALUE OF SC IS 0
5217  021640  100000                    SC                      ;CHANGE SC BIT
5218  021642                    7$:
5219
5220  021642  043737  004740  004636    BIC     @#ATTENT,@#SAVERE+24    ;CLEAR APPROPIATE ATA BITS
5221                                                                    ;FOR WORKING DRIVE IN SAVED RHAS
5222
5223                                    ;*NOW COMPARE REGISTERS AFTER THE UNLOAD COMMAND
5224                                    ;*WITH EXPECTED VALUES
5225
5226
5227  021650  004037  042512            JSR     R0,@#COMREG     ;COMPARE SAVED REGISTERS WITH
5228                                            ;PRESENT VALUE
5229  021654  004612                    SAVERE                  ;GOOD DATA SAVED IN 'SAVERE'
5230  021656  002354                    WC                      ;TEST DATA STARTING FROM 'RHWC'
5231  021660  000021                    17.                     ;17. REGISTERS TO BE COMPARED
5232  021662  021666                    3$                      ;RETURN TO 3$ ON ERROR
5233  021664  021672                    4$                      ;RETURN TO 4$ ON NO ERROR
5234
5235  021666  104023            3$:     ERROR   23              ;UNLOAD COMMAND GAVE
5236  021670  000207                    RTS     PC              ;AN ERROR
5237                                            ;GOOD DATA GIVES WHAT SHOULD
5238                                            ;BE THERE
```

```
                                                        ;RECEIVED DATA GIVES WHAT WAS
                                                        ;THERE AFTER UNLOAD COMMAND

022672                                  4$:
022672  104401  021700                          TYPE    ,98$            ;;TYPE ASCIZ STRING
022676  000406                          BR      97$             ;;GET OVER THE ASCIZ
                                        ;;98$:  .ASCIZ  <15><12>/ON DRIVE /
022714                                  97$:
022714  013746  004716                          MOV     @#UNIT,-(SP)    ;GET UNIT UNDER TEST
022720  104405                          TYPDS
022722  104401  021730                          TYPE    ,100$           ;;TYPE ASCIZ STRING
022726  000444                          BR      99$             ;;GET OVER THE ASCIZ
                                        ;;100$: .ASCIZ  <15><12>/IF STANDBY LIT HIT STANDBY OR START - AFTER HEAD LOAD HIT CONTI
022040                                  99$:
022040  005037  047342                          CLR     @#PRITEM        ;CLEAR PREVIOUS ERROR NUMBER
022044  000000                          HALT                    ;WAIT FOR CONTINUE
022046  004737  041562                          JSR     PC,@#CHECK      ;CHECKS THAT DVA,RDY,MOL & DPR = 1
022052  000240                          NOP                     ;CHECKS THAT ALL OTHER BITS = 0
022054  000240                          NOP                     ;THERE IS NO HALT IF IT FAILS - IT
022056  000240                          NOP                     ;IS USED IN THE MIDDLE OF A TEST


                                        ;*SET VV IN RHDS1 AFTER RESET FROM ACTUATING
                                        ;*THE STANDBY SWITCH AND CYCLING UP (MOL = 1)


022060  013746  002460                  MOV     @#PKACK,-(SP)   ;GET READY TO MOVE COMMAND
022064  052716  000001                  BIS     #GO,(SP)        ;GET READY TO SET GO
                                                                ;WITHOUT INTERRUPT ENABLE
022070  012677  160204                  MOV     (SP)+,@RHCS1    ;GO WITH
                                                                 ;22 IN RHCS1 FOR PACK ACKNOWLEDGE
                                                                ;WITH INTERRUPT DISABLED


022074  011100                  MOV     @R1,R0          ;SAVE RHCS1 DURING ABOVE OPERATION
022076  011305                  MOV     @R3,R5          ;SAVE RHDS1 DURING ABOVE OPERATION

022100  104413                  WAT                     ;WAIT FOR VV BIT TO SET
022102  002322                  RHDS1                   ;WAIT FOR RHDS1 REGISTER
022104  000100                  VV                      ;WAIT FOR VV BIT IN RHDS1 REGISTER
022106  000001                  1.                      ;ALLOW 10 MICRO SECONDS
022110  000001                  1.                      ;VV MUST SET BETWEEN
                                                        ;00 AND 20 MICRO SECONDS
```

```
                                ;;*****************************************************************
                                ;*TEST 31       OFFSET AND RETURN TO CENTER LINE COMMAND

                                ;*      THIS TESTS TWO COMMANDS: (1)OFFSET, (2)RETURN-TO-CENTER-LINE

                                ;*      ALL POSSIBLE REGISTERS ARE FILLED WITH ONES (EXCEPT RHOF)
                                ;*      AND AN OFFSET IS GIVEN
                                ;*      ALL REGISTERS ARE COMPARED, ONLY ATA SHOULD SET
                                ;*      ALL OTHER REGISTERS SHOULD REMAIN UNCHANGED

                                ;*      THEN A RETURN-TO-CENTER-LINE IS GIVEN
                                ;*      ALL REGISTERS ARE COMPARED ONLY ATA SHOULD SET
                                ;*      AND RHOF SHOULD CLEAR (EXCEPT HCI,ECI,FMT22)
                                ;*      ALL OTHER REGISTERS SHOULD REMAIN UNCHANGED

                                ;*      THE ABOVE PROCESS IS REPEATED FOR OFFSET REGISTER
                                ;*      VALUES OF 1 TO 377 IE. 377 TIMES

                                ;;*****************************************************************
022112  000004                  TST31:  SCOPE
022114  012737  022166  001106          MOV     #1$,$LPADR      ;;SET SCOPE LOOP ADDRESS
022122  012706  001000                  MOV     #STACK,SP       ;RESET STACK
022126  012737  000031  004604          MOV     #31,@#TSTNM     ;SAVE TEST NUMBER

022134  004737  041524                  JSR     PC,@#CLDISK     ;SET R1-RHCS1, R2-RHCS2
                                                                ;R3-RHDS1, R4-RHER1
                                                                ;GIVE RH-11 INITIALIZE
                                                                ;SETUP UNIT NUMBER
022140  004737  041562                  JSR     PC,@#CHECK      ;CHECK THAT DVA,RDY,MOL,DPR,DRY = 1
                                                                ;AND THAT NO STATUS BITS ARE STUCK = 1
022144  104401  067033                  TYPE    ,CPHALT         ;CANNOT CONTINUE TESTS IF THEY AREN'T
022150  000000                          HALT                    ;STOP
022152  112737  000001  004610          MOVB    #1,@#OFSTVL     ;SET OFFSET VALUE TO 1
022160  112737  000034  004611          MOVB    #34,@#OFSTVL+1  ;SET HCI,ECI,FMT22


022166                          1$:

022166  004737  041524                  JSR     PC,@#CLDISK     ;SET R1-RHCS1, R2-RHCS2
                                                                ;R3-RHDS1, R4-RHER1
                                                                ;GIVE RH-11 INITIALIZE
                                                                ;SETUP UNIT NUMBER
022172  004737  041604                  JSR     PC,@#CHECKT     ;CHECK DVA,RDY,MOL,DPR & VV = 1
022176  000240                          NOP                     ;CHECK THAT ALL OTHER BITS = 0
022200  000240                          NOP                     ;UNLIKE THE OTHER STATUS BIT TESTS,
022202  000240                          NOP                     ;THERE IS NO HALT IF IT FAILS - IT IS
                                                                ;USED IN THE MIDDLE OF A SINGLE TEST
022204  012700  002272                  MOV     #RHWC,R0        ;ADDR. OF ADDR OF RHWC IN R0

022210  012730  177777                  MOV     #177777,@(R0)+  ;LOAD 177777 INTO RHWC


022214  012730  177777                  MOV     #177777,@(R0)+  ;LOAD 177777 INTO RHBA


022220  052730  000010                  BIS     #10,@(R0)+      ;LOAD 10 INTO RHCS2
```

```
5339
5340  022224  012730  001400                  MOV     #1400,@(R0)+    ;LOAD 1400 INTO RHCS1
5341
5342
5343  022230  012730  000000                  MOV     #0,@(R0)+       ;LOAD 0 INTO RHER1
5344
5345
5346  022234  012730  177777                  MOV     #177777,@(R0)+  ;LOAD 177777 INTO RHDST
5347
5348
5349  022240  012730  000000                  MOV     #0,@(R0)+       ;LOAD 0 INTO RHER2
5350
5351                                          ;*THE OFFSET REGISTER WILL BE INCREMENTED FROM 0 TO 377
5352
5353  022244  013730  004610                  MOV     @#OFSTVL,@(R0)+ ;SET OFFSET REGISTER
5354
5355  022250  012730  177777                  MOV     #177777,@(R0)+  ;LOAD 177777 INTO RHCA
5356
5357  022254  012730  000000                  MOV     #0,@(R0)+       ;LOAD 0 INTO RHER3
5358
5359                                          ;*NOW SET BITS IN RHAS FOR ALL DRIVES PRESENT
5360
5361  022260  010046                          MOV     R0,-(SP)        ;;PUSH R0 ON STACK
5362  022262  011446                          MOV     @R4,-(SP)       ;SAVE RHER1 TO REINSTATE LATER
5363  022264  011246                          MOV     @R2,-(SP)       ;SAVE RHCS2 TO BE REINSTATED
5364                                                                  ;AFTER ALL ATA BITS HAVE BEEN SET
5365  022266  013700  004742                  MOV     @#TOTALAT,R0    ;GET DRIVES PRESENT
5366  022272  005012                          CLR     @R2             ;CLEAR RHCS2 AND CARRY
5367  022274  012705  000010                  MOV     #8.,R5          ;COUNTER
5368  022300  006000                  82$:    ROR     R0              ;GET BIT INTO CARRY
5369  022302  103002                          BCC     83$             ;BRANCH IF NO UNIT ON THIS BIT
5370  022304  012714  177777                  MOV     #-1,@R4         ;MOVE INTO ERROR REGISTER
5371                                                                  ;TO SET ATA BIT
5372  022310  005212                  83$:    INC     @R2             ;INCREMENT RHCS2 TO NEXT UNIT
5373  022312  005305                          DEC     R5              ;COUNT
5374  022314  001371                          BNE     82$             ;BRANCH IF 8 NOT DONE
5375  022316  012612                          MOV     (SP)+,@R2       ;REINSTATE RHCS2
5376  022320  012614                          MOV     (SP)+,@R4       ;REINSTATE RHER1
5377  022322  012600                          MOV     (SP)+,R0        ;;POP STACK INTO R0
5378  022324  005720                          TST     (R0)+           ;GET OVER PHAS IN R0
5379
5380
5381  022326  012730  177776                  MOV     #177776,@(R0)+  ;LOAD 177776 INTO RHMR
5382
5383  022332  013777  002454  157740          MOV     @#OFSETC,@RHCS1 ;GET READY FOR OFSETC
5384
5385
5386                                          ;*NOW SAVE REGISTERS FOR COMPARISON AFTER OFFSET
5387
5388  022340  004037  041672                  JSR     R0,@#SAVER      ;SAVE REGISTERS
5389  022344  002272                          RHWC                    ;RHWC IS THE FIRST REGISTER SAVED
5390  022346  004612                          SAVERE                  ;STARTING ADDRESS OF WHERE
5391                                                                  ;THE REGISTERS ARE SAVED
5392  022350  000022                          18.                     ;NUMBER OF REGISTERS
5393                                                                  ;SAVED = 18.
5394
```

```
5395  022352  013777  004606  157706          MOV     @#RP4VEC,@RPVEC ;SET RP04 VECTOR ADDRESS
5396                                                                  ;TO 'TIME1' IF P-CLOCK IS PRESENT
5397                                                                  ;OR TO 'TIME2' IF P-CLOCK IS NOT PRESENT
5398                                                                  ;'TIME' WILL ONLY SAVE
5399                                                                  ;CURRENT CYLINDER ADDRESS
5400                                                                  ;AND LOOK AHEAD REGISTERS
5401
5402
5403  022360  013746  002454                  MOV     @#OFSETC,-(SP)  ;GET READY TO MOVE COMMAND
5404  022364  052716  000101                  BIS     #GO!IE,(SP)     ;GET READY TO SET 'GO' AND
5405                                                                  ;ENABLE INTERRUPT
5406  022370  012677  157704                  MOV     (SP)+,@RHCS1    ;GO WITH
5407                                                                   ;14 IN RHCS1 FOR OFSET
5408                                                                  ;WITH INTERRUPT ENABLED
5409  022374  011100                          MOV     @R1,R0          ;SAVE RHCS1 DURING ABOVE OPERATION
5410  022376  011305                          MOV     @R3,R5          ;SAVE RHDS1 DURING ABOVE OPERATION
5411
5412
5413  022400  104413                          WAT                     ;WAIT FOR DRY BIT TO SET
5414  022402  002322                          RHDS1                   ;WAIT FOR RHDS1 REGISTER
5415  022404  000200                          DRY                     ;WAIT FOR DRY BIT IN RHDS1 REGISTER
5416  022406  001750                          1000.                   ;ALLOW 10000 MICRO SECONDS
5417  022410  000454                          300.                    ;DRY MUST SET BETWEEN
5418                                                                  ;7000 AND 13000 MICRO SECONDS
5419
5420                                  ;*COMPARE CONTENTS OF RHCS1 AND RHDS1 ALREADY SAVED IN
5421                                  ;*R0 AND R5 IMMEDIATELY AFTER GO
5422
5423  022412  013746  002454                  MOV     @#OFSETC,-(SP)  ;SAVE COMMAND
5424  022416  052716  004301                  BIS     #DVA!GO!IE!RDY,(SP)     ;INCLUDE DVA!GO!IE!RDY
5425  022422  005737  004722                  TST     @#NUNIT         ;ARE THERE MORE THAN ONE UNIT
5426  022426  001413                          BEQ     87$             ;BRANCH IF ONLY ONE UNIT
5427  022430  010037  004760                  MOV     R0,@#TMP4       ;GET RHCS1
5428  022434  042737  177677  004760          BIC     #^CIE,@#TMP4    ;KEEP IE BIT
5429  022442  042716  000100                  BIC     #IE,(SP)        ;CLEAR IE IN GOOD DATA
5430  022446  053716  004760                  BIS     @#TMP4,(SP)     ;GET IE AS IS
5431  022452  052716  100000                  BIS     #SC,(SP)        ;SET SC IN RHCS1
5432  022456                          87$:
5433  022456  011637  001124                  MOV     (SP),@#$GDDAT   ;SAVE FOR PRINTOUT
5434  022462  022600                          CMP     (SP)+,R0        ;DURING ABOVE OPERATION ONLY DVA!GO!IE!RDY
5435                                                                  ;AND COMMAND SHOULD BE SET
5436  022464  001405                          BEQ     86$             ;BRANCH IF GOOD
5437  022466  010037  001126                  MOV     R0,@#$BDDAT     ;BAD DATA
5438  022472  010137  004600                  MOV     R1,@#REGADR     ;FAILING REGISTER RHCS1
5439  022476  104021                          ERROR   21              ;DURING ABOVE OPERATION ONLY
5440                                                                  ;COMMAND AND DVA!GO!IE!RDY SHOULD BE SET
5441  022500  012746  030500          86$:    MOV     #PIP!MOL!DPR!VV,-(SP)   ;SAVE BITS SET DURING OPERATION IN RHDS1
5442  022504  011637  001124                  MOV     (SP),@#$GDDAT   ;SAVE FOR PRINTOUT
5443  022510  022605                          CMP     (SP)+,R5        ;DURING ABOVE OPERATION ONLY PIP!MOL!DPR!VV
5444                                                                  ;SHOULD BE SET
5445  022512  001405                          BEQ     88$             ;BRANCH IF GOOD
5446  022514  010537  001126                  MOV     R5,@#$BDDAT     ;BAD DATA
5447  022520  010337  004600                  MOV     R3,@#REGADR     ;FAILING REGISTER RHDS1
5448  022524  104063                          ERROR   63              ;DURING ABOVE OPERATION ONLY
5449                                                                  ;PIP!MOL!DPR!VV SHOULD BE SET
5450  022526                          88$:
```

```
                                        ;*NOW CHANGE SAVED REGISTERS TO EXPECTED VALUES

022526  004037  042404                  JSR     R0,@#CHREG      ;CHANGE BITS IN SAVED REGISTER
022532  002300                          RHCS1                   ;CHANGE RHCS1 REGISTER

022534  000001                          1                       ;1 BIT/BITS TO BE CHANGED
022536  000001                          1                       ;NEW VALUE OF SC IS 1
022540  100000                          SC                      ;CHANGE SC BIT

022542  004037  042404                  JSR     R0,@#CHREG      ;CHANGE BITS IN SAVED REGISTER
022546  002322                          RHDS1                   ;CHANGE RHDS1 REGISTER

022550  000001                          1                       ;1 BIT/BITS TO BE CHANGED
022552  000001                          1                       ;NEW VALUE OF ATA IS 1
022554  100000                          ATA                     ;CHANGE ATA BIT

022556  053737  004740  004636          BIS     @#ATTENT,@#SAVERE+24    ;SET APPROPRIATE 'ATA' BITS
                                                                ;FOR WORKING DRIVE IN
                                                                ;SAVED RHAS LOACTION

                                        ;*NOW COMPARE REGISTERS AFTER AN OFSET COMMAND

022564  004037  042512                  JSR     R0,@#COMREG     ;COMPARE SAVED REGISTERS WITH
                                                                ;PRESENT VALUE
022570  004612                          SAVERE                  ;GOOD DATA SAVED IN 'SAVERE'
022572  002354                          WC                      ;TEST DATA STARTING FROM 'RHWC'
022574  000022                          18.                     ;18. REGISTERS TO BE COMPARED
022576  022602                          2$                      ;RETURN TO 2$ ON ERROR
022600  022606                          3$                      ;RETURN TO 3$ ON NO ERROR

022602  104024                  2$:     ERROR   24              ;OFFSET COMMAND CAUSED AN ERROR
022604  000207                          RTS     PC              ;GOOD DATA IS WHAT SHOULD BE THERE
                                                                ;RECEIVED DATA GIVES WHAT WAS THERE
                                                                ;AFTER AN OFFSET COMMAND

022606  013777  004740  157502  3$:     MOV     @#ATTENT,@RHAS  ;CLEAR WORKING DRIVE ATTENTION


                                        ;*NOW A RETURN TO CENTER LINE COMMAND WILL BE GIVEN
                                        ;;**************************************************************

022614  004737  041604                  JSR     PC,@#CHECKT     ;CHECK DVA,RDY,MOL,DPR & VV = 1
022620  000240                          NOP                     ;CHECK THAT ALL OTHER BITS = 0
022622  000240                          NOP                     ;UNLIKE THE OTHER STATUS BIT TESTS,
022624  000240                          NOP                     ;THERE IS NO HALT IF IT FAILS - IT IS
                                                                ;USED IN THE MIDDLE OF A SINGLE TEST
022626  013777  002456  157444          MOV     @#RETCL,@RHCS1  ;GET READY FOR RETCL
                                                                ;RETURN TO CENTER LINE WITH 16 IN RHCS1

                                        ;*NOW REGISTERS ARE SAVED FOR COMPARISON AFTER COMMAND
```

```
5507
5508  022634  004037  041672          JSR     R0,@#SAVER      ;SAVE REGISTERS
5509  022640  002272                  RHWC                    ;RHWC IS THE FIRST REGISTER SAVED
5510  022642  004612                  SAVERE                  ;STARTING ADDRESS OF WHERE
5511                                                          ;THE REGISTERS ARE SAVED
5512  022644  000022                  18.                     ;NUMBER OF REGISTERS
5513                                                          ;SAVED = 18.
5514
5515  022646  013777  004606  157412  MOV     @#RP4VEC,@RPVEC ;SET RP04 VECTOR ADDRESS
5516                                                          ;TO 'TIME1' IF P-CLOCK IS PRESENT
5517                                                          ;OR TO 'TIME2' IF P-CLOCK IS NOT PRESENT
5518                                                          ;'TIME' WILL ONLY SAVE
5519                                                          ;CURRENT CYLINDER ADDRESS
5520                                                          ;AND LOOK AHEAD REGISTERS
5521
5522
5523  022654  013746  002456          MOV     @#RETCL,-(SP)   ;GET READY TO MOVE COMMAND
5524  022660  052716  000101          BIS     #GO!IE,(SP)     ;GET READY TO SET 'GO' AND
5525                                                          ;ENABLE INTERRUPT
5526  022664  012677  157410          MOV     (SP)+,@RHCS1    ;GO WITH
5527                                                           ;16 IN RHCS1 FOR RETURN TO CENTER LINE
5528                                                          ;WITH INTERRUPT ENABLED
5529  022670  011100                  MOV     @R1,R0          ;SAVE RHCS1 DURING ABOVE OPERATION
5530  022672  011305                  MOV     @R3,R5          ;SAVE RHDS1 DURING ABOVE OPERATION
5531
5532
5533  022674  104413                  WAT                     ;WAIT FOR DRY BIT TO SET
5534  022676  002322                  RHDS1                   ;WAIT FOR RHDS1 REGISTER
5535  022700  000200                  DRY                     ;WAIT FOR DRY BIT IN RHDS1 REGISTER
5536  022702  001750                  1000.                   ;ALLOW 10000 MICRO SECONDS
5537  022704  001750                  1000.                   ;DRY MUST SET BETWEEN
5538                                                          ;00 AND 20000 MICRO SECONDS
5539
5540                                  ;*COMPARE CONTENTS OF RHCS1 AND RHDS1 ALREADY SAVED IN
5541                                  ;*R0 AND R5 IMMEDIATELY AFTER GO
5542
5543  022706  013746  002456          MOV     @#RETCL,-(SP)   ;SAVE COMMAND
5544  022712  052716  004301          BIS     #DVA!GO!IE!RDY,(SP)     ;INCLUDE DVA!GO!IE!RDY
5545  022716  005737  004722          TST     @#NUNIT         ;ARE THERE MORE THAN ONE UNIT
5546  022722  001413                  BEQ     90$             ;BRANCH IF ONLY ONE UNIT
5547  022724  010037  004760          MOV     R0,@#TMP4       ;GET RHCS1
5548  022730  042737  177677  004760  BIC     #^CIE,@#TMP4    ;KEEP IE BIT
5549  022736  042716  000100          BIC     #IE,(SP)        ;CLEAR IE IN GOOD DATA
5550  022742  053716  004760          BIS     @#TMP4,(SP)     ;GET IE AS IS
5551  022746  052716  100000          BIS     #SC,(SP)        ;SET SC IN RHCS1
5552  022752                  90$:
5553  022752  011637  001124          MOV     (SP),@#$GDDAT   ;SAVE FOR PRINTOUT
5554  022756  022600                  CMP     (SP)+,R0        ;DURING ABOVE OPERATION ONLY DVA!GO!IE!RDY
5555                                                          ;AND COMMAND SHOULD BE SET
5556  022760  001405                  BEQ     89$             ;BRANCH IF GOOD
5557  022762  010037  001126          MOV     R0,@#$BDDAT     ;BAD DATA
5558  022766  010137  004600          MOV     R1,@#REGADR     ;FAILING REGISTER RHCS1
5559  022772  104021                  ERROR   21              ;DURING ABOVE OPERATION ONLY
5560                                                          ;COMMAND AND DVA!GO!IE!RDY SHOULD BE SET
5561  022774  012746  030500  89$:    MOV     #PIP!MOL!DPR!VV,-(SP)   ;SAVE BITS SET DURING OPERATION IN RHDS1
5562  023000  011637  001124          MOV     (SP),@#$GDDAT   ;SAVE FOR PRINTOUT
```

```
5563  023004  022605                          CMP     (SP)+,R5        ;DURING ABOVE OPERATION ONLY PIP!MOL!DPR!VV
5564                                                                  ;SHOULD BE SET
5565  023006  001405                          BEQ     91$             ;BRANCH IF GOOD
5566  023010  010537  001126                  MOV     R5,@#$BDDAT     ;BAD DATA
5567  023014  010337  004600                  MOV     R3,@#REGADR     ;FAILING REGISTER RHDS1
5568  023020  104063                          ERROR   63              ;DURING ABOVE OPERATION ONLY
5569                                                                  ;PIP!MOL!DPR!VV SHOULD BE SET
5570  023022                          91$:
5571
5572                                          ;*NOW CHANGE SAVED REGISTER TO EXPECTED VALUE
5573
5574
5575  023022  004037  042404                  JSR     R0,@#CHREG      ;CHANGE BITS IN SAVED REGISTER
5576  023026  002300                          RHCS1                   ;CHANGE RHCS1 REGISTER
5577
5578  023030  000001                          1                       ;1 BIT/BITS TO BE CHANGED
5579  023032  000001                          1                       ;NEW VALUE OF SC IS 1
5580  023034  100000                          SC                      ;CHANGE SC BIT
5581
5582  023036  004037  042404                  JSR     R0,@#CHREG      ;CHANGE BITS IN SAVED REGISTER
5583  023042  002322                          RHDS1                   ;CHANGE RHDS1 REGISTER
5584
5585  023044  000001                          1                       ;1 BIT/BITS TO BE CHANGED
5586  023046  000001                          1                       ;NEW VALUE OF ATA IS 1
5587  023050  100000                          ATA                     ;CHANGE ATA BIT
5588
5589  023052  053737  004740  004636          BIS     @#ATTENT,@#SAVERE+24    ;SET APPROPRIATE 'ATA' BITS
5590                                                                  ;FOR WORKING DRIVE IN
5591                                                                  ;SAVED RHAS LOACTION
5592  023060  004037  041426                  JSR     R0,@#FILLRE     ;MOV BIT15!HCI!ECI!FMT22 INTO SAVED RHOF
5593  023064  002310                          RHOF                    ;SAVED REGISTER TO CHANGE
5594  023066  116000                          BIT15!HCI!ECI!FMT22                     ;DATA
5595
5596                                          ;*NOW COMPARE REGISTERS AFTER RETURN-TO-CENTER-LINE
5597
5598  023070  004037  042512                  JSR     R0,@#COMREG     ;COMPARE SAVED REGISTERS WITH
5599                                                                  ;PRESENT VALUE
5600  023074  004612                          SAVERE                  ;GOOD DATA SAVED IN 'SAVERE'
5601  023076  002354                          WC                      ;TEST DATA STARTING FROM 'RHWC'
5602  023100  000022                          18.                     ;18. REGISTERS TO BE COMPARED
5603  023102  023106                          4$                      ;RETURN TO 4$ ON ERROR
5604  023104  023112                          5$                      ;RETURN TO 5$ ON NO ERROR
5605  023106  104025          4$:             ERROR   25              ;RETURN TO CENTER-LINE
5606  023110  000207                          RTS     PC              ;COMMAND CAUSED AN ERROR
5607                                                                  ;GOOD DATA HAS WHAT SHOULD
5608                                                                  ;BE THERE
5609                                                                  ;RECEIVED DATA HAS WHAT WAS
5610                                                                  ;THERE AFTER COMMAND
5611
5612  023112                          5$:
5613
5614  023112  004737  041524                  JSR     PC,@#CLDISK     ;SET R1-RHCS1, R2-RHCS2
5615                                                                  ;R3-RHDS1, R4-RHER1
5616                                                                  ;GIVE RH-11 INITIALIZE
5617                                                                  ;SETUP UNIT NUMBER
5618  023116  105237  004610                  INCB    @#OFSTVL        ;GET NEXT OFFSET VALUE
```

```
 5619  023122  132737  000100  004610          BITB    #100,@#OFSTVL   ;SEE IF UNUSED BIT 6  IS ON
 5620  023130  001403                          BEQ     7$              ;NO SO DO SOME MORE
 5621  023132  062737  000100  004610          ADD     #100,@#OFSTVL   ;YES SO BY-PASS IT
 5622  023140  105737  004610          7$:     TSTB    @#OFSTVL        ;IF ZERO ALL COMBINATIONS ARE
 5623                                                                  ;COMPLETE
 5624  023144  001001                          BNE     6$              ;BRANCH IF 377 NOT DONE
 5625  023146  000402                          BR      TST32   ;       BRANCH TO NEXT TEST
 5626  023150  000137  022166          6$:     JMP     @#1$            ;JUMP BECAUSE 377 NOT DONE
 5627
```

```
                                ;;*****************************************************************
                                ;*TEST 32       OFFSET COMMAND
                                ;
                                ;*      THIS TEST WILL ONLY GIVE REPEATED OFFSETS
                                ;
                                ;;*****************************************************************
023154  000004                  TST32:  SCOPE
023156  012737  000764  004756          MOV     # 500.,@#TMP1   ;COUNTER
023164                          1$:
023164  012706  001000          MOV     #STACK,SP       ;RESET STACK
023170  012737  000032  004604  MOV     #32,@#TSTNM     ;SAVE TEST NUMBER
                                ;
023176  004737  041524          JSR     PC,@#CLDISK     ;SET R1-RHCS1, R2-RHCS2
                                                        ;R3-RHDS1, R4-RHER1
                                                        ;GIVE RH-11 INITIALIZE
                                                        ;SETUP UNIT NUMBER
023202  004737  041604          JSR     PC,@#CHECKT     ;CHECK DVA,RDY,MOL,DPR,DRY,VV = 1
                                                        ;AND THAT NO STATUS BITS ARE STUCK = 1
023206  104401  067033          TYPE    ,CPHALT         ;CANNOT CONTINUE TESTING IF ANY OF
                                                        ;THE FIRST SET OF BITS DON'T = 1
023212  000000                  HALT                    ;STOP
023214  004037  041510          JSR     R0,@#OFSET      ;OFSET
023220  000260                  260                     ;260 IN OFSET REGISTER
                                                        ;OFSET -1200 MICRO INCHES
                                ;
023222  013746  002454          MOV     @#OFSETC,-(SP)  ;GET READY TO MOVE COMMAND
023226  052716  000101          BIS     #GO!IE,(SP)     ;GET READY TO SET 'GO' AND
                                                        ;ENABLE INTERRUPT
023232  012677  157042          MOV     (SP)+,@RHCS1    ;GO WITH
                                                         ;14 IN RHCS1 FOR OFSET
                                                        ;WITH INTERRUPT ENABLED
                                ;
023236  104413                  WAT                     ;WAIT FOR DRY BIT TO SET
023240  002322                  RHDS1                   ;WAIT FOR RHDS1 REGISTER
023242  000200                  DRY                     ;WAIT FOR DRY BIT IN RHDS1 REGISTER
023244  000536                  350.                    ;ALLOW 3500 MICRO SECONDS
023246  000536                  350.                    ;DRY MUST SET BETWEEN
                                                        ;00 AND 7000 MICRO SECONDS
023250  032777  040000  157044  BIT     #ERR,@RHDS1     ;IS ERR SET?
023256  001417                  BEQ     2$              ;NO
023260  104026                  ERROR   26              ;REPEATED OFFSETS CAUSED AN ERROR
023262  000004                  SCOPE
                                ;
023264  004737  041524          JSR     PC,@#CLDISK     ;SET R1-RHCS1, R2-RHCS2
                                                        ;R3-RHDS1, R4-RHER1
                                                        ;GIVE RH-11 INITIALIZE
                                                        ;SETUP UNIT NUMBER
                                ;
023270  013746  002426          MOV     @#RECALI,-(SP)  ;GET READY TO MOVE COMMAND
023274  052716  000101          BIS     #GO!IE,(SP)     ;GET READY TO SET 'GO' AND
                                                        ;ENABLE INTERRUPT
023300  012677  156774          MOV     (SP)+,@RHCS1    ;GO WITH
                                                        ;6 IN RHCS1 FOR RECALIBRATE
                                                        ;WITH INTERRUPT ENABLED
```

```
 5684  023304  104413                        WAT                     ;WAIT FOR DRY BIT TO SET
 5685  023306  002322                        RHDS1                   ;WAIT FOR RHDS1 REGISTER
 5686  023310  000200                        DRY                     ;WAIT FOR DRY BIT IN RHDS1 REGISTER
 5687  023312  060650                        25000.                  ;ALLOW 250000 MICRO SECONDS
 5688  023314  060650                        25000.                  ;DRY MUST SET BETWEEN
 5689                                                                ;00 AND 500000 MICRO SECONDS
 5690  023316  005337  004756          2$:   DEC     @#TMP1          ;COUNT DOWN
 5691  023322  001401                        BEQ     TST33   ;       BRANCH IF DONE
 5692  023324  000717                        BR      1$              ;GO BACK AND DO IT AGIEN
 5693
```

```
                                .SBTTL  READ/WRITE TESTS USING MEDIA
                                ;;*****************************************************************
                                ;*TEST 33       WRITE/READ HEADER AND DATA (0'S)

                                ;*      WRITE HEADER AND DATA CYLINDER 0, FORMAT 16 BITS PER WORD
                                ;*      TRACK 0, SECTOR 0, KEYS=0, NUMBER OF WORDS 256 WORDS
                                ;*      OF 0
                                ;*      THEN READ HEADER AND DATA FOR ABOVE.
                                ;*      WRITE FROM BUFFER AND READ INTO BUFFER ARE FILLED WITH
                                ;*      10000,0,0,0, AND 256 OF 0
                                ;*      THE WRITE COMMAND IS THEN LOADED INTO THE REGISTERS EXCEPT
                                ;*      THE GO BIT, AND ALL THE REGISTERS ARE SAVED
                                ;*      THEN GO IS GIVEN FOR WRITE HEADER AND DATA
                                ;*
                                ;*      THEN ALL REGISTERS ARE COMPARED TO CHECK FOR IMPROPER CHANGED
                                ;*      THEN WRITE FROM BUFFER IS CHECKED TO SEE THAT NOTHING CHANGED
                                ;*
                                ;*
                                ;*      NOW FOR THE READ COMMAND READ INTO BUFFER IS FILLED
                                ;*      WITH ALL ONES, COMMAND IS LOADED INTO REGISTERS EXCEPT
                                ;*      GO BIT AND ALL REGISTERS ARE SAVED
                                ;*      GO IS GIVEN FOR THE READ COMMAND
                                ;*
                                ;*      ALL REGISTERS ARE CHECKED FOR IMPROPER CHANGE
                                ;*      THEN THE READ DATA IS COMPARED.

                                ;;*****************************************************************
023326  000004                  TST33:  SCOPE
023330  012706  001000                  MOV     #STACK,SP       ;RESET STACK
023334  012737  000033  004604          MOV     #33,@#TSTNM     ;SAVE TEST NUMBER

023342  004737  041524          JSR     PC,@#CLDISK     ;SET R1-RHCS1, R2-RHCS2
                                                        ;R3-RHDS1, R4-RHER1
                                                        ;GIVE RH-11 INITIALIZE
                                                        ;SETUP UNIT NUMBER

                                ;*FILL WRITE FROM BUFFER WITH HEADER

023346  004037  041350          JSR     R0,@#FLHEAD     ;SAVE HEADER DATA IN WRFROM
023352  002470          -       WRFROM                  ;LOCATION WHERE SAVED
023354  000004                  4                       ;NUMBER OF WORDS SAVED
023356  010000                  10000                   ;FIRST DATA WORD
023360  000000                  0                       ;SECOND DATA WORD
023362  000000                  0                       ;THIRD DATA WORD
023364  000000                  0                       ;FOURTH DATA WORD

                                ;*FILL WRITE FROM BUFFER WITH DATA

023366  004037  041374          JSR     R0,@#CLAREA     ;CLEAR 256. WORDS, FROM WRFROM+10
023372  002500                  WRFROM+10                       ;STARTING FROM WRFROM+10
023374  000400                  256.                    ;256. WORDS
023376  000000                  0                       ;FILL WITH 0


                                ;*NOW READ INTO BUFFER WILL BE FILLED WITH SAME DATA
```

```
                                        ;*AS WRITE FROM BUFFER SO THAT AFTER A WRITE COMPARISONS
                                        ;*CAN BE MADE TO MAKE SURE THAT WRITE DID NOT
                                        ;*CHANGE WRITE FROM BUFFER


023400  004037  041350                  JSR     R0,@#FLHEAD     ;SAVE HEADER DATA IN REINTO
023404  003534                          REINTO                  ;LOCATION WHERE SAVED
023406  000004                          4                       ;NUMBER OF WORDS SAVED
023410  010000                          10000                   ;FIRST DATA WORD
023412  000000                          0                       ;SECOND DATA WORD
023414  000000                          0                       ;THIRD DATA WORD
023416  000000                          0                       ;FOURTH DATA WORD
023420  004037  041374                  JSR     R0,@#CLAREA     ;CLEAR 256. WORDS, FROM REINTO+10
023424  003544                          REINTO+10                       ;STARTING FROM REINTO+10
023426  000400                          256.                    ;256. WORDS
023430  000000                          0                       ;FILL WITH 0


                                        ;*NOW THE WRITE HEADER AND DATA COMMAND WILL BE FILLED


023432  004037  043476                  JSR     R0,@#RUN        ;SETUP TO RUN FOR DATA COMMAND
023436  000000                          0                       ;CYLINDER 0
023440     000                  .BYTE   0                       ;SECTOR 0
023441     000                  .BYTE   0                       ;TRACK 0
023442  177374                          -256.-4                 ;WORD COUNT (DATA) = 256. +
                                                                ;4 HEADER WORDS
023444  002470                          WRFROM                  ;BUS ADDRESS
                                                                ;STARTING ADDRESS OF DATA
                                                                ;BUFFER = WRFROM
023446  000000                          0                       ;DO NOT INHIBIT BUS ADDRESS INCREMENT
023450  010000                          FMT22                   ;16 BITS PER WORD FORMAT
                                                                ;DO NOT INHIBIT ECC CORRECTION
                                                                ;DO NOT INHIBIT HEADER COMPARE
023452  002444                          WRIFOR                  ;GET READY TO DO A WRIFOR
                                                        ;WRITE HEADER AND DATA WITH 62 IN RHCS1


                                        ;*NOW SAVE REGISTERS FOR COMPARISON AFTER WRITE HEADER AND DATA

023454  004037  041672                  JSR     R0,@#SAVER      ;SAVE REGISTERS
023460  002272                          RHWC                    ;RHWC IS THE FIRST REGISTER SAVED
023462  004612                          SAVERE                  ;STARTING ADDRESS OF WHERE
                                                                ;THE REGISTERS ARE SAVED
023464  000021                          17.                     ;NUMBER OF REGISTERS
                                                                ;SAVED = 17.

023466  004737  041604                  JSR     PC,@#CHECKT     ;CHECK DVA,RDY,MOL,DPR,DRY,VV = 1
                                                                ;AND THAT NO STATUS BITS ARE STUCK = 1
023472  104401  067033                  TYPE    ,CPHALT         ;CANNOT CONTINUE TESTING IF ANY OF
                                                                ;THE FIRST SET OF BITS DON'T = 1
023476  000000                          HALT                    ;STOP

023500  013777  004606  156560          MOV     @#RP4VEC,@RPVEC ;SET RP04 VECTOR ADDRESS
                                                                ;TO 'TIME1' IF P-CLOCK IS PRESENT
                                                                ;OR TO 'TIME2' IF P-CLOCK IS NOT PRESENT
```

```
                                                        ;'TIME' WILL ONLY SAVE
                                                        ;CURRENT CYLINDER ADDRESS
                                                        ;AND LOOK AHEAD REGISTERS

023506  013746  002444                  MOV     @#WRIFOR,-(SP)  ;GET READY TO MOVE COMMAND
023512  052716  000101                  BIS     #GO!IE,(SP)     ;GET READY TO SET 'GO' AND
                                                        ;ENABLE INTERRUPT
023516  012677  156556                  MOV     (SP)+,@RHCS1    ;GO WITH
                                                         ;62 IN RHCS1 FOR WRITE HEADER AND DATA
                                                        ;WITH INTERRUPT ENABLED
023522  011100                          MOV     @R1,R0          ;SAVE RHCS1 DURING ABOVE OPERATION
023524  011305                          MOV     @R3,R5          ;SAVE RHDS1 DURING ABOVE OPERATION

                                        ;*ONE REVOLUTION=16670 MICRO SEC, ONE SECTOR = 760 MICRO SEC


023526  104413                          WAT                     ;WAIT FOR RDY BIT TO SET
023530  002300                          RHCS1                   ;WAIT FOR RHCS1 REGISTER
023532  000200                          RDY                     ;WAIT FOR RDY BIT IN RHCS1 REGISTER
023534  001614                          908.                    ;ALLOW 9080 MICRO SECONDS
023536  001507                          839.                    ;RDY MUST SET BETWEEN
                                                        ;690 AND 17470 MICRO SECONDS

                                        ;*COMPARE CONTENTS OF RHCS1 AND RHDS1 ALREADY SAVED IN
                                        ;*R0 AND R5 IMMEDIATELY AFTER GO

023540  013746  002444                  MOV     @#WRIFOR,-(SP)  ;SAVE COMMAND
023544  052716  004101                  BIS     #IE!GO!DVA,(SP) ;INCLUDE IE!GO!DVA
023550  011637  001124                  MOV     (SP),@#$GDDAT   ;SAVE FOR PRINTOUT
023554  022600                          CMP     (SP)+,R0        ;DURING ABOVE OPERATION ONLY IE:GO:DVA
                                                        ;AND COMMAND SHOULD BE SET
023556  001405                          BEQ     64$             ;BRANCH IF GOOD
023560  010037  001126                  MOV     R0,@#$BDDAT     ;BAD DATA
023564  010137  004600                  MOV     R1,@#REGADR     ;FAILING REGISTER RHCS1
023570  104021                          ERROR   21              ;DURING ABOVE OPERATION ONLY
                                                        ;COMMAND AND IE!GO!DVA SHOULD BE SET
023572  012746  010500          64$:    MOV     #MOL!DPR!VV,-(SP)       ;SAVE BITS SET DURING OPERATION IN RHDS1
023576  011637  001124                  MOV     (SP),@#$GDDAT   ;SAVE FOR PRINTOUT
023602  022605                          CMP     (SP)+,R5        ;DURING ABOVE OPERATION ONLY MOL!DPR!VV
                                                        ;SHOULD BE SET
023604  001405                          BEQ     66$             ;BRANCH IF GOOD
023606  010537  001126                  MOV     R5,@#$BDDAT     ;BAD DATA
023612  010337  004600                  MOV     R3,@#REGADR     ;FAILING REGISTER RHDS1
023616  104063                          ERROR   63              ;DURING ABOVE OPERATION ONLY
                                                        ;MOL!DPR!VV SHOULD BE SET
023620                          66$:

                                        ;*NOW CHANGE SAVED REGISTERS TO EXPECTED VALUES


023620  004037  041426                  JSR     R0,@#FILLRE     ;MOV 0 INTO SAVED RHWC
023624  002272                          RHWC                    ;SAVED REGISTER TO CHANGE
023626  000000                          0                       ;DATA
023630  004037  041426                  JSR     R0,@#FILLRE     ;MOV WRFROM+<260.*2> INTO SAVED RHBA
023634  002274                          RHBA                    ;SAVED REGISTER TO CHANGE
023636  003500                          WRFROM+<260.*2>                 ;DATA
```

```
 5862  023640  004037  041426                JSR     R0,@#FILLRE     ;MOV 1 INTO SAVED RHDST
 5863  023644  002304                        RHDST                   ;SAVED REGISTER TO CHANGE
 5864  023646  000001                        1                       ;DATA
 5865
 5866                                        ;*NOW COMPARE REGISTERS BEFORE WRITE HEADER AND DATA
 5867                                        ;*WITH REGISTERS AFTER COMMAND
 5868
 5869
 5870  023650  004037  042512                JSR     R0,@#COMREG     ;COMPARE SAVED REGISTERS WITH
 5871                                                                ;PRESENT VALUE
 5872  023654  004612                        SAVERE                  ;GOOD DATA SAVED IN 'SAVERE'
 5873  023656  002354                        WC                      ;TEST DATA STARTING FROM 'RHWC'
 5874  023660  000021                        17.                     ;17. REGISTERS TO BE COMPARED
 5875  023662  023666                        1$                      ;RETURN TO 1$ ON ERROR
 5876  023664  023672                        2$                      ;RETURN TO 2$ ON NO ERROR
 5877
 5878  023666  104027              1$:       ERROR   27              ;WRITE HEADER AND DATA
 5879  023670  000207                        RTS     PC              ;CAUSED IMPROPER REGISTER
 5880                                                                ;CHANGE
 5881                                                                ;GOOD DATA GIVES WHAT SHOULD
 5882                                                                ;BE THERE
 5883                                                                ;RECEIVED DATA GIVES WHAT
 5884                                                                ;WAS THERE AFTER COMMAND
 5885
 5886                                        ;*NOW WRITE FROM BUFFER WILL BE CHECKED TO SEE THAT
 5887                                        ;*NOTHING GOT CHANGED
 5888
 5889  023672                        2$:
 5890
 5891  023672  004037  043542                JSR     R0,@#COMPAR     ;COMPARE TWO BLOCKS OF MEMORY
 5892  023676  003534                        REINTO                  ;GOOD DATA STARTS FROM REINTO
 5893  023700  002470                        WRFROM                  ;TEST DATA STARTS FROM WRFROM
 5894  023702  000404                        260.                    ;260. WORDS TO BE COMPARED
 5895  023704  023710                        3$                      ;RETURN TO 3$ ON ERROR
 5896  023706  023714                        4$                      ;RETURN TO 4$ ON NO ERROR
 5897
 5898
 5899  023710  104030              3$:       ERROR   30              ;WRITE HEADER AND DATA
 5900  023712  000207                        RTS     PC              ;CHANGED WRITE FROM BUFFER
 5901
 5902                                        ;*NOW A READ HEADER AND DATA COMMAND WILL BE GIVEN
 5903                                        ;*READ INTO BUFFER IS FILLED WITH ONES
 5904
 5905  023714                        4$:
 5906
 5907  023714  004737  041524                JSR     PC,@#CLDISK     ;SET R1-RHCS1, R2-RHCS2
 5908                                                                ;R3-RHDS1, R4-RHER1
 5909                                                                ;GIVE RH-11 INITIALIZE
 5910                                                                ;SETUP UNIT NUMBER
 5911  023720  004037  041374                JSR     R0,@#CLAREA     ;CLEAR 260. WORDS, FROM REINTO
 5912  023724  003534                        REINTO                  ;STARTING FROM REINTO
 5913  023726  000404                        260.                    ;260. WORDS
 5914  023730  177777                        -1                      ;FILL WITH -1
 5915
 5916
 5917                                        ;*NOW FILL COMMAND
```

```
5918
5919
5920  023732  004037  043476                JSR     R0,@#RUN        ;SETUP TO RUN FOR DATA COMMAND
5921  023736  000000                        0                       ;CYLINDER 0
5922  023740     000              .BYTE     0                       ;SECTOR 0
5923  023741     000              .BYTE     0                       ;TRACK 0
5924  023742  177374                        -256.-4                 ;WORD COUNT (DATA) = 256. +
5925                                                                ;4 HEADER WORDS
5926  023744  003534                        REINTO                  ;BUS ADDRESS
5927                                                                ;STARTING ADDRESS OF DATA
5928                                                                ;BUFFER = REINTO
5929  023746  000000                        0                       ;DO NOT INHIBIT BUS ADDRESS INCREMENT
5930  023750  014000                        ECI!FMT22                       ;16 BITS PER WORD FORMAT
5931                                                                ;INHIBIT ECC CORRECTION
5932                                                                ;DO NOT INHIBIT HEADER COMPARE
5933  023752  002450                        REFOR                   ;GET READY TO DO A REFOR
5934                                                                ;READ HEADER AND DATA WITH 72 IN RHCS1
5935
5936
5937                                        ;*NOW SAVE REGISTERS FOR COMPARISON AFTER READ HEADER AND DATA
5938
5939  023754  004037  041672                JSR     R0,@#SAVER      ;SAVE REGISTERS
5940  023760  002272                        RHWC                    ;RHWC IS THE FIRST REGISTER SAVED
5941  023762  004612                        SAVERE                  ;STARTING ADDRESS OF WHERE
5942                                                                ;THE REGISTERS ARE SAVED
5943  023764  000022                        18.                     ;NUMBER OF REGISTERS
5944                                                                ;SAVED = 18.
5945
5946  023766  004737  041604                JSR     PC,@#CHECKT     ;CHECK DVA,RDY,MOL,DPR,DRY,VV = 1
5947                                                                ;AND THAT NO STATUS BITS ARE STUCK = 1
5948  023772  104401  067033                TYPE    ,CPHALT         ;CANNOT CONTINUE TESTING IF ANY OF
5949                                                                ;THE FIRST SET OF BITS DON'T = 1
5950  023776  000000                        HALT                    ;STOP
5951
5952  024000  013777  004606  156260        MOV     @#RP4VEC,@RPVEC ;SET RP04 VECTOR ADDRESS
5953                                                                ;TO 'TIME1' IF P-CLOCK IS PRESENT
5954                                                                ;OR TO 'TIME2' IF P-CLOCK IS NOT PRESENT
5955                                                                ;'TIME' WILL ONLY SAVE
5956                                                                ;CURRENT CYLINDER ADDRESS
5957                                                                ;AND LOOK AHEAD REGISTERS
5958
5959
5960  024006  013746  002450                MOV     @#REFOR,-(SP)   ;GET READY TO MOVE COMMAND
5961  024012  052716  000101                BIS     #GO!IE,(SP)     ;GET READY TO SET 'GO' AND
5962                                                                ;ENABLE INTERRUPT
5963  024016  012677  156256                MOV     (SP)+,@RHCS1    ;GO WITH
5964                                                                  ;72 IN RHCS1 FOR READ DATA
5965                                                                ;WITH INTERRUPT ENABLED
5966  024022  011100                        MOV     @R1,R0          ;SAVE RHCS1 DURING ABOVE OPERATION
5967  024024  011305                        MOV     @R3,R5          ;SAVE RHDS1 DURING ABOVE OPERATION
5968
5969
5970  024026  104413                        WAT                     ;WAIT FOR RDY BIT TO SET
5971  024030  002300                        RHCS1                   ;WAIT FOR RHCS1 REGISTER
5972  024032  000200                        RDY                     ;WAIT FOR RDY BIT IN RHCS1 REGISTER
5973  024034  001614                        908.                    ;ALLOW 9080 MICRO SECONDS
```

```
                                        839.                    ;RDY MUST SET BETWEEN
                                                                ;690 AND 17470 MICRO SECONDS

                                        ;*COMPARE CONTENTS OF RHCS1 AND RHDS1 ALREADY SAVED IN
                                        ;*R0 AND R5 IMMEDIATELY AFTER GO

024040  013746  002450                  MOV     @#REFOR,-(SP)   ;SAVE COMMAND
024044  052716  004101                  BIS     #IE!GO!DVA,(SP) ;INCLUDE IE!GO!DVA
024050  011637  001124                  MOV     (SP),@#$GDDAT   ;SAVE FOR PRINTOUT
024054  022600                          CMP     (SP)+,R0        ;DURING ABOVE OPERATION ONLY IE!GO!DVA
                                                                ;AND COMMAND SHOULD BE SET
024056  001405                          BEQ     67$             ;BRANCH IF GOOD
024060  010037  001126                  MOV     R0,@#$BDDAT     ;BAD DATA
024064  010137  004600                  MOV     R1,@#REGADR     ;FAILING REGISTER RHCS1
024070  104021                          ERROR   21              ;DURING ABOVE OPERATION ONLY
                                                                ;COMMAND AND IE!GO!DVA SHOULD BE SET
024072  012746  010500          67$:    MOV     #MOL!DPR!VV,-(SP)       ;SAVE BITS SET DURING OPERATION IN RHDS1
024076  011637  001124                  MOV     (SP),@#$GDDAT   ;SAVE FOR PRINTOUT
024102  022605                          CMP     (SP)+,R5        ;DURING ABOVE OPERATION ONLY MOL!DPR!VV
                                                                ;SHOULD BE SET
024104  001405                          BEQ     69$             ;BRANCH IF GOOD
024106  010537  001126                  MOV     R5,@#$BDDAT     ;BAD DATA
024112  010337  004600                  MOV     R3,@#REGADR     ;FAILING REGISTER RHDS1
024116  104063                          ERROR   63              ;DURING ABOVE OPERATION ONLY
                                                                ;MOL!DPR!VV SHOULD BE SET
024120                          69$:

                                        ;*CHANGE SAVED REGISTERS TO EXPECTED VALUES

024120  004037  041426                  JSR     R0,@#FILLRE     ;MOV 0 INTO SAVED RHWC
024124  002272                          RHWC                    ;SAVED REGISTER TO CHANGE
024126  000000                          0                       ;DATA
024130  004037  041426                  JSR     R0,@#FILLRE     ;MOV REINTO+<260.*2> INTO SAVED RHBA
024134  002274                          RHBA                    ;SAVED REGISTER TO CHANGE
024136  004544                          REINTO+<260.*2>                 ;DATA
024140  004037  041426                  JSR     R0,@#FILLRE     ;MOV 1 INTO SAVED RHDST
024144  002304                          RHDST                   ;SAVED REGISTER TO CHANGE
024146  000001                          1                       ;DATA

                                        ;*COMPARE REGISTER BEFORE READ HEADER AND DATA
                                        ;*WITH REGISTERS AFTER COMMAND


024150  004037  042512                  JSR     R0,@#COMREG     ;COMPARE SAVED REGISTERS WITH
                                                                ;PRESENT VALUE
024154  004612                          SAVERE                  ;GOOD DATA SAVED IN 'SAVERE'
024156  002354                          WC                      ;TEST DATA STARTING FROM 'RHWC'
024160  000022                          18.                     ;18. REGISTERS TO BE COMPARED
024162  024166                          5$                      ;RETURN TO 5$ ON ERROR
024164  024172                          6$                      ;RETURN TO 6$ ON NO ERROR

024166  104031                  5$:     ERROR   31              ;READ HEADER AND DATA CAUSED
024170  000207                          RTS     PC              ;IMPROPER REGISTER CHANGE
                                                                ;GOOD DATA GIVES WHAT SHOULD
                                                                ;BE THERE
                                                                ;RECEIVED DATA GIVES WHAT WAS
```

```
                                                        ;THERE AFTER COMMAND

                                ;*NOW READ INTO BUFFER WILL BE CHECKED TO SEE
                                ;*THE READ WAS GOOD

024172                      6$:

024172  004037  043542          JSR     R0,@#COMPAR     ;COMPARE TWO BLOCKS OF MEMORY
024176  002470                  WRFROM                  ;GOOD DATA STARTS FROM WRFROM
024200  003534                  REINTO                  ;TEST DATA STARTS FROM REINTO
024202  000404                  260.                    ;260. WORDS TO BE COMPARED
024204  024210                  7$                      ;RETURN TO 7$ ON ERROR
024206  024214                  10$                     ;RETURN TO 10$ ON NO ERROR

024210  104032              7$: ERROR   32              ;WRITE HEADER AND DATA
024212  000207                  RTS     PC              ;FOLLOWED BY A READ HEADER
                                                        ;AND DATA GAVE A READ ERROR
                                                        ;ERROR MAY BE IN READ OR WRITE
024214                      10$:
```

```
6051
6052                                    ;;*****************************************************************
6053                                    ;*TEST 34       READ DATA (0'S)
6054
6055                                    ;*      THIS TEST READS DATA WRITTEN BY THE PREVIOUS TEST
6056                                    ;*      THE WRITE FROM BUFFER IS FILLED WITH ALL 0S
6057                                    ;*      THE COMMAND IS FILLED, THEN ALL REGISTERS SAVED
6058                                    ;*      THEN READ DATA COMMAND IS GIVEN
6059                                    ;*      ALL REGISTERS ARE COMPARED FOR PROPER VALUES
6060                                    ;*      READ DATA INTO 'READ INTO' BUFFER IS CHECKED
6061
6062                                    ;;*****************************************************************
6063  024214  000004                    TST34:  SCOPE
6064  024216  012706  001000                    MOV     #STACK,SP       ;RESET STACK
6065  024222  012737  000034  004604            MOV     #34,@#TSTNM     ;SAVE TEST NUMBER
6066
6067  024230  004737  041524                    JSR     PC,@#CLDISK     ;SET R1-RHCS1, R2-RHCS2
6068                                                                    ;R3-RHDS1, R4-RHER1
6069                                                                    ;GIVE RH-11 INITIALIZE
6070                                                                    ;SETUP UNIT NUMBER
6071
6072                                    ;*FILL WRITE FROM BUFFER WITH EXPECTED DATA
6073
6074  024234  004037  041374                    JSR     R0,@#CLAREA     ;CLEAR 256. WORDS, FROM WRFROM
6075  024240  002470                            WRFROM                  ;STARTING FROM WRFROM
6076  024242  000400                            256.                    ;256. WORDS
6077  024244  000000                            0                       ;FILL WITH 0
6078
6079
6080                                    ;*NOW THE READ DATA COMMAND WILL BE FILLED
6081
6082
6083  024246  004037  043476                    JSR     R0,@#RUN        ;SETUP TO RUN FOR DATA COMMAND
6084  024252  000000                            0                       ;CYLINDER 0
6085  024254     000                .BYTE   0                       ;SECTOR 0
6086  024255     000                .BYTE   0                       ;TRACK 0
6087  024256  177400                            -256.                   ;WORD COUNT = 256.
6088  024260  003534                            REINTO                  ;BUS ADDRESS
6089                                                                    ;STARTING ADDRESS OF DATA
6090                                                                    ;BUFFER = REINTO
6091  024262  000000                            0                       ;DO NOT INHIBIT BUS ADDRESS INCREMENT
6092  024264  014000                            EC1!FMT22                       ;16 BITS PER WORD FORMAT
6093                                                                    ;INHIBIT ECC CORRECTION
6094                                                                    ;DO NOT INHIBIT HEADER COMPARE
6095  024266  002446                            READAT                  ;GET READY TO DO A READAT
6096                                                            ;READ DATA WITH 70 IN RHCS1
6097
6098
6099                                    ;*NOW SAVE REGISTERS FOR COMPARISON AFTER READ DATA COMMAND
6100
6101  024270  004037  041672                    JSR     R0,@#SAVER      ;SAVE REGISTERS
6102  024274  002272                            RHWC                    ;RHWC IS THE FIRST REGISTER SAVED
6103  024276  004612                            SAVERE                  ;STARTING ADDRESS OF WHERE
6104                                                                    ;THE REGISTERS ARE SAVED
6105  024300  000022                            18.                     ;NUMBER OF REGISTERS
6106                                                                    ;SAVED = 18.
```

```
6107
6108  024302  004737  041604                JSR     PC,@#CHECKT     ;CHECK DVA,RDY,MOL,DPR,DRY,VV = 1
6109                                                                  ;AND THAT NO STATUS BITS ARE STUCK = 1
6110  024306  104401  067033                TYPE    ,CPHALT         ;CANNOT CONTINUE TESTING IF ANY OF
6111                                                                  ;THE FIRST SET OF BITS DON'T = 1
6112  024312  000000                        HALT                    ;STOP
6113
6114  024314  013777  004606  155744        MOV     @#RP4VEC,@RPVEC ;SET RP04 VECTOR ADDRESS
6115                                                                  ;TO 'TIME1' IF P-CLOCK IS PRESENT
6116                                                                  ;OR TO 'TIME2' IF P-CLOCK IS NOT PRESENT
6117                                                                  ;'TIME' WILL ONLY SAVE
6118                                                                  ;CURRENT CYLINDER ADDRESS
6119                                                                  ;AND LOOK AHEAD REGISTERS
6120
6121
6122  024322  013746  002446                MOV     @#READAT,-(SP)  ;GET READY TO MOVE COMMAND
6123  024326  052716  000101                BIS     #GO!IE,(SP)     ;GET READY TO SET 'GO' AND
6124                                                                  ;ENABLE INTERRUPT
6125  024332  012677  155742                MOV     (SP)+,@RHCS1    ;GO WITH
6126                                                                   ;70 IN RHCS1 FOR READ DATA
6127                                                                  ;WITH INTERRUPT ENABLED
6128  024336  011100                        MOV     @R1,R0          ;SAVE RHCS1 DURING ABOVE OPERATION
6129  024340  011305                        MOV     @R3,R5          ;SAVE RHDS1 DURING ABOVE OPERATION
6130
6131
6132  024342  104413                        WAT                     ;WAIT FOR RDY BIT TO SET
6133  024344  002300                        RHCS1                   ;WAIT FOR RHCS1 REGISTER
6134  024346  000200                        RDY                     ;WAIT FOR RDY BIT IN RHCS1 REGISTER
6135  024350  001614                        908.                    ;ALLOW 9080 MICRO SECONDS
6136  024352  001507                        839.                    ;RDY MUST SET BETWEEN
6137                                                                  ;690 AND 17470 MICRO SECONDS
6138
6139                                        ;*COMPARE CONTENTS OF RHCS1 AND RHDS1 ALREADY SAVED IN
6140                                        ;*R0 AND R5 IMMEDIATELY AFTER GO
6141
6142  024354  013746  002446                MOV     @#READAT,-(SP)  ;SAVE COMMAND
6143  024360  052716  004101                BIS     #IE!GO!DVA,(SP) ;INCLUDE IE!GO!DVA
6144  024364  011637  001124                MOV     (SP),@#$GDDAT   ;SAVE FOR PRINTOUT
6145  024370  022600                        CMP     (SP)+,R0        ;DURING ABOVE OPERATION ONLY IE!GO!DVA
6146                                                                  ;AND COMMAND SHOULD BE SET
6147  024372  001405                        BEQ     64$             ;BRANCH IF GOOD
6148  024374  010037  001126                MOV     R0,@#$BDDAT     ;BAD DATA
6149  024400  010137  004600                MOV     R1,@#REGADR     ;FAILING REGISTER RHCS1
6150  024404  104021                        ERROR   21              ;DURING ABOVE OPERATION ONLY
6151                                                                  ;COMMAND AND IE!GO!DVA SHOULD BE SET
6152  024406  012746  010500        64$:    MOV     #MOL!DPR!VV,-(SP)       ;SAVE BITS SET DURING OPERATION IN RHDS1
6153  024412  011637  001124                MOV     (SP),@#$GDDAT   ;SAVE FOR PRINTOUT
6154  024416  022605                        CMP     (SP)+,R5        ;DURING ABOVE OPERATION ONLY MOL!DPR!VV
6155                                                                  ;SHOULD BE SET
6156  024420  001405                        BEQ     66$             ;BRANCH IF GOOD
6157  024422  010537  001126                MOV     R5,@#$BDDAT     ;BAD DATA
6158  024426  010337  004600                MOV     R3,@#REGADR     ;FAILING REGISTER RHDS1
6159  024432  104063                        ERROR   63              ;DURING ABOVE OPERATION ONLY
6160                                                                  ;MOL!DPR!VV SHOULD BE SET
6161  024434                        66$:
6162
```

```
                                        ;*NOW CHANGE SAVED REGISTERS TO EXPECTED VALUES
                                        ;
024434  004037  041426                  JSR     R0,@#FILLRE     ;MOV 0 INTO SAVED RHWC
024440  002272                          RHWC                    ;SAVED REGISTER TO CHANGE
024442  000000                          0                       ;DATA
024444  004037  041426                  JSR     R0,@#FILLRE     ;MOV REINTO+<256.*2> INTO SAVED RHBA
024450  002274                          RHBA                    ;SAVED REGISTER TO CHANGE
024452  004534                          REINTO+<256.*2>                 ;DATA
024454  004037  041426                  JSR     R0,@#FILLRE     ;MOV 1 INTO SAVED RHDST
024460  002304                          RHDST                   ;SAVED REGISTER TO CHANGE
024462  000001                          1                       ;DATA
                                        ;
                                        ;*NOW COMPARE REGISTERS BEFORE READ DATA WITH
                                        ;*AFTER COMMAND
                                        ;
024464  004037  042512                  JSR     R0,@#COMREG     ;COMPARE SAVED REGISTERS WITH
                                                                ;PRESENT VALUE
024470  004612                          SAVERE                  ;GOOD DATA SAVED IN 'SAVERE'
024472  002354                          WC                      ;TEST DATA STARTING FROM 'RHWC'
024474  000022                          18.                     ;18. REGISTERS TO BE COMPARED
024476  024502                          1$                      ;RETURN TO 1$ ON ERROR
024500  024506                          2$                      ;RETURN TO 2$ ON NO ERROR
                                        ;
024502  104033                  1$:     ERROR   33              ;READ DATA CAUSED IMPROPER REGISTER
024504  000207                          RTS     PC              ;CHANGE
                                                                ;GOOD DATA GIVES WHAT SHOULD BE THERE
                                                                ;RECEIVED DATA GIVES WHAT WAS THERE AFTER COMMAND
                                        ;*NOW READ INTO BUFFER WILL BE CHECKED TO SEE THAT READ
                                        ;*WAS GOOD
                                        ;
024506                          2$:
                                        ;
024506  004037  043542                  JSR     R0,@#COMPAR     ;COMPARE TWO BLOCKS OF MEMORY
024512  002470                          WRFROM                  ;GOOD DATA STARTS FROM WRFROM
024514  003534                          REINTO                  ;TEST DATA STARTS FROM REINTO
024516  000400                          256.                    ;256. WORDS TO BE COMPARED
024520  024524                          3$                      ;RETURN TO 3$ ON ERROR
024522  024530                          4$                      ;RETURN TO 4$ ON NO ERROR
                                        ;
                                        ;
024524  104034                  3$:     ERROR   34              ;READ DATA COMMAND
024526  000207                          RTS     PC              ;READ INCORRECTLY
                                        ;
024530                          4$:
```

```
6208
6209                                    ;;*****************************************************************
6210                                    ;*TEST 35       WRITE/READ DATA (1'S & 125252)
6211
6212                                    ;*      THIS TEST GIVES A WRITE DATA COMMAND FRO CYLINDER 0
6213                                    ;*      TRACK 0, SECTOR 0, KEYS 0, 200 WORDS OF ALL ONES
6214                                    ;*      THIS SECTOR IS FORMATED BY PREVIOUS TEST
6215                                    ;*      THEN READ DATA COMMAND IS GIVEN FOR 256 WORDS IN
6216                                    ;*      SAME CYLINDER, TRACK, SECTOR, KEYS
6217                                    ;*      WRITE FROM BUFFER AND READ INTO BUFFER ARE FILLED WITH
6218                                    ;*      200 ONES AND 56 125252
6219                                    ;*      THE WRITE DATA COMMAND IS LOADED EXCEPT GO AND IE
6220                                    ;*      ALL REGISTERS ARE SAVED AND THEN GO IS GIVEN TO WRITE DATA
6221                                    ;*
6222                                    ;*
6223                                    ;*      THEN ALL REGISTERS ARE COMPARED TO CHECK FOR IMPROPER
6224                                    ;*      CHANGE, THEN WRITE FROM BUFFER IS CHECKED FOR NO CHANGE
6225                                    ;*
6226                                    ;*      THEN READ INTO BUFFER IS FILLED WITH 200 OF ZEROS
6227                                    ;*      AND 56 OF 377, WRITE FROM BUFFER IS FILLED WITH 200 ONES
6228                                    ;*      AND 56 OF 377.
6229                                    ;*      THE COMMAND EXCEPT GO IS LOADED, ALL REGISTERS ARE SAVED
6230                                    ;*      GO IS GIVEN
6231                                    ;*
6232                                    ;*
6233                                    ;*      ALL REGISTERS ARE CHECKED
6234                                    ;*      READ DATA IS CHECKED
6235
6236                                    ;;*****************************************************************
6237  024530  000004                    TST35:  SCOPE
6238  024532  012706  001000                    MOV     #STACK,SP       ;RESET STACK
6239  024536  012737  000035  004604            MOV     #35,@#TSTNM     ;SAVE TEST NUMBER
6240
6241  024544  004737  041524                    JSR     PC,@#CLDISK     ;SET R1-RHCS1, R2-RHCS2
6242                                                                    ;R3-RHDS1, R4-RHER1
6243                                                                    ;GIVE RH-11 INITIALIZE
6244                                                                    ;SETUP UNIT NUMBER
6245
6246                                    ;*NOW FILL WRITE FROM BUFFER -200 OF 1'S AND 56 OF 125252
6247
6248  024550  004037  041374                    JSR     R0,@#CLAREA     ;CLEAR 200. WORDS, FROM WRFROM
6249  024554  002470                    WRFROM                          ;STARTING FROM WRFROM
6250  024556  000310                    200.                            ;200. WORDS
6251  024560  177777                    -1                              ;FILL WITH -1
6252
6253  024562  004037  041374                    JSR     R0,@#CLAREA     ;CLEAR 56. WORDS, FROM WRFROM+<200.*2>
6254  024566  003310                    WRFROM+<200.*2>                         ;STARTING FROM WRFROM+<200.*2>
6255  024570  000070                    56.                             ;56. WORDS
6256  024572  125252                    125252                          ;FILL WITH 125252
6257
6258
6259                                    ;*NOW READ INTO BUFFER WILL BE FILLED WITH SAME DATA AS
6260                                    ;*WRITE FROM BUFFER SO THAT AFTER A WRITE COMPARISONS
6261                                    ;*CAN BE MADE TO DETERMINE THAT WRITE DID NOT CHANGE BUFFER
6262
6263  024574  004037  041374                    JSR     R0,@#CLAREA     ;CLEAR 200. WORDS, FROM REINTO
```

```
                                        REINTO                  ;STARTING FROM REINTO
                                        200.                    ;200. WORDS
                                        -1                      ;FILL WITH -1

                                        JSR     R0,@#CLAREA     ;CLEAR 56. WORDS, FROM REINTO+<200.*2>
                                        REINTO+<200.*2>                 ;STARTING FROM REINTO+<200.*2>
                                        56.                     ;56. WORDS
                                        125252                  ;FILL WITH 125252


                                        ;*NOW WRITE DATA COMMAND WILL BE LOADED


                                        JSR     R0,@#RUN        ;SETUP TO RUN FOR DATA COMMAND
                                        0                       ;CYLINDER 0
                                .BYTE   0                       ;SECTOR 0
                                .BYTE   0                       ;TRACK 0
                                        -200.                   ;WORD COUNT = 200.
                                        WRFROM                  ;BUS ADDRESS
                                                                ;STARTING ADDRESS OF DATA
                                                                ;BUFFER = WRFROM
                                        0                       ;DO NOT INHIBIT BUS ADDRESS INCREMENT
                                        FMT22                   ;16 BITS PER WORD FORMAT
                                                                ;DO NOT INHIBIT ECC CORRECTION
                                                                ;DO NOT INHIBIT HEADER COMPARE
                                        WRIDAT                  ;GET READY TO DO A WRIDAT
                                                        ;WRITE DATA WITH 60 IN RHCS1


                                        ;*NOW SAVE REGISTERS FOR COMPARISON AFTER WRITE DATA

                                        JSR     R0,@#SAVER      ;SAVE REGISTERS
                                        RHWC                    ;RHWC IS THE FIRST REGISTER SAVED
                                        SAVERE                  ;STARTING ADDRESS OF WHERE
                                                                ;THE REGISTERS ARE SAVED
                                        18.                     ;NUMBER OF REGISTERS
                                                                ;SAVED = 18.

                                        JSR     PC,@#CHECKT     ;CHECK DVA,RDY,MOL,DPR,DRY,VV = 1
                                                                ;AND THAT NO STATUS BITS ARE STUCK = 1
                                        TYPE    ,CPHALT         ;CANNOT CONTINUE TESTING IF ANY OF
                                                                ;THE FIRST SET OF BITS DON'T = 1
                                        HALT                    ;STOP

                                        MOV     @#RP4VEC,@RPVEC ;SET RP04 VECTOR ADDRESS
                                                                ;TO 'TIME1' IF P-CLOCK IS PRESENT
                                                                ;OR TO 'TIME2' IF P-CLOCK IS NOT PRESENT
                                                                ;'TIME' WILL ONLY SAVE
                                                                ;CURRENT CYLINDER ADDRESS
                                                                ;AND LOOK AHEAD REGISTERS


                                        MOV     @#WRIDAT,-(SP)  ;GET READY TO MOVE COMMAND
                                        BIS     #GO!IE,(SP)     ;GET READY TO SET 'GO' AND
                                                                ;ENABLE INTERRUPT
                                        MOV     (SP)+,@RHCS1    ;GO WITH
```

```
                                                        ;60 IN RHCS1 FOR WRITE DATA
                                                       ;WITH INTERRUPT ENABLED
024710  011100                      MOV     @R1,R0       ;SAVE RHCS1 DURING ABOVE OPERATION
024712  011305                      MOV     @R3,R5       ;SAVE RHDS1 DURING ABOVE OPERATION

                                    ;*ONE REVOLUTION = 16670 MICRO SEC, ONE SECTOR=760 MICRO SEC


024714  104413                      WAT                  ;WAIT FOR RDY BIT TO SET
024716  002300                      RHCS1                ;WAIT FOR RHCS1 REGISTER
024720  000200                      RDY                  ;WAIT FOR RDY BIT IN RHCS1 REGISTER
024722  001614                      908.                 ;ALLOW 9080 MICRO SECONDS
024724  001507                      839.                 ;RDY MUST SET BETWEEN
                                                         ;690 AND 17470 MICRO SECONDS

                                    ;*COMPARE CONTENTS OF RHCS1 AND RHDS1 ALREADY SAVED IN
                                    ;*R0 AND R5 IMMEDIATELY AFTER GO

024726  013746  002442              MOV     @#WRIDAT,-(SP)  ;SAVE COMMAND
024732  052716  004101              BIS     #IE!GO!DVA,(SP) ;INCLUDE IE!GO!DVA
024736  011637  001124              MOV     (SP),@#$GDDAT   ;SAVE FOR PRINTOUT
024742  022600                      CMP     (SP)+,R0        ;DURING ABOVE OPERATION ONLY IE!GO!DVA
                                                         ;AND COMMAND SHOULD BE SET
024744  001405                      BEQ     64$             ;BRANCH IF GOOD
024746  010037  001126              MOV     R0,@#$BDDAT     ;BAD DATA
024752  010137  004600              MOV     R1,@#REGADR     ;FAILING REGISTER RHCS1
024756  104021                      ERROR   21              ;DURING ABOVE OPERATION ONLY
                                                         ;COMMAND AND IE!GO!DVA SHOULD BE SET
024760  012746  010500      64$:    MOV     #MOL!DPR!VV,-(SP)       ;SAVE BITS SET DURING OPERATION IN RHDS1
024764  011637  001124              MOV     (SP),@#$GDDAT   ;SAVE FOR PRINTOUT
024770  022605                      CMP     (SP)+,R5        ;DURING ABOVE OPERATION ONLY MOL!DPR!VV
                                                         ;SHOULD BE SET
024772  001405                      BEQ     66$             ;BRANCH IF GOOD
024774  010537  001126              MOV     R5,@#$BDDAT     ;BAD DATA
025000  010337  004600              MOV     R3,@#REGADR     ;FAILING REGISTER RHDS1
025004  104063                      ERROR   63              ;DURING ABOVE OPERATION ONLY
                                                         ;MOL!DPR!VV SHOULD BE SET
025006                      66$:

                                    ;*NOW CHANGE SAVED REGISTERS TO EXPECTED VALUE

025006  004037  041426              JSR     R0,@#FILLRE     ;MOV 0 INTO SAVED RHWC
025012  002272                      RHWC                    ;SAVED REGISTER TO CHANGE
025014  000000                      0                       ;DATA
025016  004037  041426              JSR     R0,@#FILLRE     ;MOV WRFROM+<200.*2> INTO SAVED RHBA
025022  002274                      RHBA                    ;SAVED REGISTER TO CHANGE
025024  003310                      WRFROM+<200.*2>                 ;DATA
025026  004037  041426              JSR     R0,@#FILLRE     ;MOV 1 INTO SAVED RHDST
025032  002304                      RHDST                   ;SAVED REGISTER TO CHANGE
025034  000001                      1                       ;DATA

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


025036  004037  042512              JSR     R0,@#COMREG     ;COMPARE SAVED REGISTERS WITH
```

```
                                                                ;PRESENT VALUE
025042  004612                        SAVERE                  ;GOOD DATA SAVED IN 'SAVERE'
025044  002354                        WC                      ;TEST DATA STARTING FROM 'RHWC'
025046  000022                        18.                     ;18. REGISTERS TO BE COMPARED
025050  025054                        1$                      ;RETURN TO 1$ ON ERROR
025052  025060                        2$                      ;RETURN TO 2$ ON NO ERROR

025054  104035              1$:       ERROR   35              ;WRITE DATA COMMAND CAUSED
025056  000207                        RTS     PC              ;IMPROPER REGISTER CHANGE
                                                                ;GOOD DATA GIVES WHAT SHOULD
                                                                ;BE
                                                                ;RECEIVED DATA GIVES WHAT WAS
                                                                ;THERE AFTER COMMAND

                                    ;*NOW WRITE FROM BUFFER WILL BE CHECKED FOR NO CHANGE

025060              2$:

025060  004037  043542              JSR     R0,@#COMPAR     ;COMPARE TWO BLOCKS OF MEMORY
025064  003534                      REINTO                  ;GOOD DATA STARTS FROM REINTO
025066  002470                      WRFROM                  ;TEST DATA STARTS FROM WRFROM
025070  000400                      256.                    ;256. WORDS TO BE COMPARED
025072  025076                      3$                      ;RETURN TO 3$ ON ERROR
025074  025102                      4$                      ;RETURN TO 4$ ON NO ERROR

025076  104036              3$:     ERROR   36              ;WRITE DATA COMMAND CHANGED
025100  000207                      RTS     PC              ;WRITE FROM BUFFER

                                    ;*NOW A READ DATA COMMAND WILL BE GIVEN

                                    ;*FILL READ INTO BUFFER WITH 200 ZEROS AND 56 OF 377

025102              4$:

025102  004737  041524              JSR     PC,@#CLDISK     ;SET R1-RHCS1, R2-RHCS2
                                                            ;R3-RHDS1, R4-RHER1
                                                            ;GIVE RH-11 INITIALIZE
                                                            ;SETUP UNIT NUMBER
025106  004037  041374              JSR     R0,@#CLAREA     ;CLEAR 200. WORDS, FROM REINTO
025112  003534                      REINTO                  ;STARTING FROM REINTO
025114  000310                      200.                    ;200. WORDS
025116  000000                      0                       ;FILL WITH 0

025120  004037  041374              JSR     R0,@#CLAREA     ;CLEAR 56. WORDS, FROM REINTO+<200.*2>
025124  004354                      REINTO+<200.*2>                 ;STARTING FROM REINTO+<200.*2>
025126  000070                      56.                     ;56. WORDS
025130  000377                      377                     ;FILL WITH 377

                                    ;*FILL WRITE FROM BUFFER WITH 200 ONES AND 56 OF 377

025132  004037  041374              JSR     R0,@#CLAREA     ;CLEAR 200. WORDS, FROM WRFROM
025136  002470                      WRFROM                  ;STARTING FROM WRFROM
025140  000310                      200.                    ;200. WORDS
025142  177777                      -1                      ;FILL WITH -1
```

```
6432
6433  025144  004037  041374                JSR     R0,@#CLAREA     ;CLEAR 56. WORDS, FROM WRFROM+<200.*2>
6434  025150  003310                        WRFROM+<200.*2>                 ;STARTING FROM WRFROM+<200.*2>
6435  025152  000070                        56.                     ;56. WORDS
6436  025154  000377                        377                     ;FILL WITH 377
6437
6438
6439                                        ;*NOW FILL COMMAND
6440
6441
6442  025156  004037  043476                JSR     R0,@#RUN        ;SETUP TO RUN FOR DATA COMMAND
6443  025162  000000                        0                       ;CYLINDER 0
6444  025164     000                .BYTE   0                       ;SECTOR 0
6445  025165     000                .BYTE   0                       ;TRACK 0
6446  025166  177470                        -200.                   ;WORD COUNT = 200.
6447  025170  003534                        REINTO                  ;BUS ADDRESS
6448                                                                ;STARTING ADDRESS OF DATA
6449                                                                ;BUFFER = REINTO
6450  025172  000000                        0                       ;DO NOT INHIBIT BUS ADDRESS INCREMENT
6451  025174  014000                        EC1!FMT22                       ;16 BITS PER WORD FORMAT
6452                                                                ;INHIBIT ECC CORRECTION
6453                                                                ;DO NOT INHIBIT HEADER COMPARE
6454  025176  002446                        READAT                  ;GET READY TO DO A READAT
6455                                                      ;READ DATA WITH 70 IN RHCS1
6456
6457
6458                                        ;*NOW SAVE REGISTERS FOR COMPARISON AFTER READ DATA COMMAND
6459
6460  025200  004037  041672                JSR     R0,@#SAVER      ;SAVE REGISTERS
6461  025204  002272                        RHWC                    ;RHWC IS THE FIRST REGISTER SAVED
6462  025206  004612                        SAVERE                  ;STARTING ADDRESS OF WHERE
6463                                                                ;THE REGISTERS ARE SAVED
6464  025210  000022                        18.                     ;NUMBER OF REGISTERS
6465                                                                ;SAVED = 18.
6466
6467  025212  004737  041604                JSR     PC,@#CHECKT     ;CHECK DVA,RDY,MOL,DPR,DRY,VV = 1
6468                                                                ;AND THAT NO STATUS BITS ARE STUCK = 1
6469  025216  104401  067033                TYPE    ,CPHALT         ;CANNOT CONTINUE TESTING IF ANY OF
6470                                                                ;THE FIRST SET OF BITS DON'T = 1
6471  025222  000000                        HALT                    ;STOP
6472
6473  025224  013777  004606  155034        MOV     @#RP4VEC,@RPVEC ;SET RP04 VECTOR ADDRESS
6474                                                                ;TO 'TIME1' IF P-CLOCK IS PRESENT
6475                                                                ;OR TO 'TIME2' IF P-CLOCK IS NOT PRESENT
6476                                                                ;'TIME' WILL ONLY SAVE
6477                                                                ;CURRENT CYLINDER ADDRESS
6478                                                                ;AND LOOK AHEAD REGISTERS
6479
6480
6481  025232  013746  002446                MOV     @#READAT,-(SP)  ;GET READY TO MOVE COMMAND
6482  025236  052716  000101                BIS     #GO!IE,(SP)     ;GET READY TO SET 'GO' AND
6483                                                                ;ENABLE INTERRUPT
6484  025242  012677  155032                MOV     (SP)+,@RHCS1    ;GO WITH
6485                                                                 ;70 IN RHCS1 FOR READ DATA
6486                                                                ;WITH INTERRUPT ENABLED
6487  025246  011100                        MOV     @R1,R0          ;SAVE RHCS1 DURING ABOVE OPERATION
```

```
6488  025250  011305                          MOV     @R3,R5          ;SAVE RHDS1 DURING ABOVE OPERATION
6489
6490
6491  025252  104413                          WAT                     ;WAIT FOR RDY BIT TO SET
6492  025254  002300                          RHCS1                   ;WAIT FOR RHCS1 REGISTER
6493  025256  000200                          RDY                     ;WAIT FOR RDY BIT IN RHCS1 REGISTER
6494  025260  001614                          908.                    ;ALLOW 9080 MICRO SECONDS
6495  025262  001507                          839.                    ;RDY MUST SET BETWEEN
6496                                                                  ;690 AND 17470 MICRO SECONDS
6497
6498                                          ;*COMPARE CONTENTS OF RHCS1 AND RHDS1 ALREADY SAVED IN
6499                                          ;*R0 AND R5 IMMEDIATELY AFTER GO
6500
6501  025264  013746  002446                  MOV     @#READAT,-(SP)  ;SAVE COMMAND
6502  025270  052716  004101                  BIS     #IE!GO!DVA,(SP) ;INCLUDE IE!GO!DVA
6503  025274  011637  001124                  MOV     (SP),@#$GDDAT   ;SAVE FOR PRINTOUT
6504  025300  022600                          CMP     (SP)+,R0        ;DURING ABOVE OPERATION ONLY IE!GO!DVA
6505                                                                  ;AND COMMAND SHOULD BE SET
6506  025302  001405                          BEQ     67$             ;BRANCH IF GOOD
6507  025304  010037  001126                  MOV     R0,@#$BDDAT     ;BAD DATA
6508  025310  010137  004600                  MOV     R1,@#REGADR     ;FAILING REGISTER RHCS1
6509  025314  104021                          ERROR   21              ;DURING ABOVE OPERATION ONLY
6510                                                                  ;COMMAND AND IE!GO!DVA SHOULD BE SET
6511  025316  012746  010500          67$:    MOV     #MOL!DPR!VV,-(SP)       ;SAVE BITS SET DURING OPERATION IN RHDS1
6512  025322  011637  001124                  MOV     (SP),@#$GDDAT   ;SAVE FOR PRINTOUT
6513  025326  022605                          CMP     (SP)+,R5        ;DURING ABOVE OPERATION ONLY MOL!DPR!VV
6514                                                                  ;SHOULD BE SET
6515  025330  001405                          BEQ     69$             ;BRANCH IF GOOD
6516  025332  010537  001126                  MOV     R5,@#$BDDAT     ;BAD DATA
6517  025336  010337  004600                  MOV     R3,@#REGADR     ;FAILING REGISTER RHDS1
6518  025342  104063                          ERROR   63              ;DURING ABOVE OPERATION ONLY
6519                                                                  ;MOL!DPR!VV SHOULD BE SET
6520  025344                          69$:
6521
6522                                          ;*CHANGE SAVED REGISTERS TO EXPECTED VALUES
6523
6524  025344  004037  041426                  JSR     R0,@#FILLRE     ;MOV 0 INTO SAVED RHWC
6525  025350  002272                          RHWC                    ;SAVED REGISTER TO CHANGE
6526  025352  000000                          0                       ;DATA
6527  025354  004037  041426                  JSR     R0,@#FILLRE     ;MOV REINTO+<200.*2> INTO SAVED RHBA
6528  025360  002274                          RHBA                    ;SAVED REGISTER TO CHANGE
6529  025362  004354                          REINTO+<200.*2>                 ;DATA
6530  025364  004037  041426                  JSR     R0,@#FILLRE     ;MOV 1 INTO SAVED RHDST
6531  025370  002304                          RHDST                   ;SAVED REGISTER TO CHANGE
6532  025372  000001                          1                       ;DATA
6533
6534                                          ;*COMPARE REGISTERS BEFORE READ DATA COMMAND
6535                                          ;*WITH REGISTERS AFTER COMMAND
6536
6537
6538  025374  004037  042512                  JSR     R0,@#COMREG     ;COMPARE SAVED REGISTERS WITH
6539                                                                  ;PRESENT VALUE
6540  025400  004612                          SAVERE                  ;GOOD DATA SAVED IN 'SAVERE'
6541  025402  002354                          WC                      ;TEST DATA STARTING FROM 'RHWC'
6542  025404  000022                          18.                     ;18. REGISTERS TO BE COMPARED
6543  025406  025412                          5$                      ;RETURN TO 5$ ON ERROR
```

```
6544  025410  025416                    6$                      ;RETURN TO 6$ ON NO ERROR
6545
6546  025412  104033              5$:   ERROR   33              ;READ DATA CAUSED IMPROPER
6547  025414  000207                    RTS     PC              ;REGISTER CHANGE
6548                                                            ;GOOD DATA GIVES WHAT SHOULD BE THERE
6549                                                            ;RECEIVED DATA GIVES WHAT WAS THERE
6550                                                            ;AFTER COMMAND
6551
6552                                    ;*NOW READ INTO BUFFER IS CHECKED FOR GOOD READ
6553
6554  025416                    6$:
6555
6556  025416  004037  043542           JSR     R0,@#COMPAR     ;COMPARE TWO BLOCKS OF MEMORY
6557  025422  002470                    WRFROM                  ;GOOD DATA STARTS FROM WRFROM
6558  025424  003534                    REINTO                  ;TEST DATA STARTS FROM REINTO
6559  025426  000400                    256.                    ;256. WORDS TO BE COMPARED
6560  025430  025434                    7$                      ;RETURN TO 7$ ON ERROR
6561  025432  025440                    10$                     ;RETURN TO 10$ ON NO ERROR
6562
6563
6564  025434  104034              7$:   ERROR   34              ;INCORRECT DATA AFTER
6565  025436  000207                    RTS     PC              ;WRITE DATA FOLLOWED BY A
6566                                                            ;READ DATA
6567  025440                    10$:
6568
6569
6570
```

```
                                ;;*****************************************************************
                                ;*TEST 36       WRITE/READ DATA (125252)
                                ;*
                                ;*      THIS TEST GIVES A WRITE DATA COMMAND FOR CYLINDER 0
                                ;*      TRACK 0, SECTOR 0, KEYS 0, 256 WORDS OF 125252
                                ;*      THIS SECTOR IS FORMATED BY PREVIOUS TEST
                                ;*      THEN READ DATA COMMAND IS GIVEN FOR 256 WORDS IN
                                ;*      SAME CYLINDER, TRACK, SECTOR, KEYS
                                ;*      WRITE FROM BUFFER AND READ INTO BUFFER ARE FILLED WITH
                                ;*      256 OF 125252
                                ;*      THE WRITE DATA COMMAND IS LOADED EXCEPT GO AND IE
                                ;*      ALL REGISTERS ARE SAVED AND THEN GO IS GIVEN TO WRITE DATI
                                ;*
                                ;*
                                ;*      THEN ALL REGISTERS ARE COMPARED TO CHECK FOR IMPROPER
                                ;*      CHANGE, THEN WRITE FROM BUFFER IS CHECKED FOR NO CHANGE
                                ;*
                                ;*      THEN READ INTO BUFFER IS FILLED WITH 256 OF ZEROS
                                ;*      WRITE FROM BUFFER IS FILLED WITH
                                ;*      256 OF 125252
                                ;*      THE COMMAND EXCEPT GO IS LOADED, ALL REGISTERS ARE SAVED
                                ;*      GO IS GIVEN
                                ;*
                                ;*
                                ;*      ALL REGISTERS ARE CHECKED
                                ;*      READ DATA IS CHECKED

                                ;;*****************************************************************
025440  000004                  TST36:  SCOPE
025442  012706  001000                  MOV     #STACK,SP       ;RESET STACK
025446  012737  000036  004604          MOV     #36,@#TSTNM     ;SAVE TEST NUMBER

025454  004737  041524                  JSR     PC,@#CLDISK     ;SET R1-RHCS1, R2-RHCS2
                                                                ;R3-RHDS1, R4-RHER1
                                                                ;GIVE RH-11 INITIALIZE
                                                                ;SETUP UNIT NUMBER

                                        ;*NOW FILL WRITE FROM BUFFER - 256 OF 125252

025460  004037  041374                  JSR     R0,@#CLAREA     ;CLEAR 256. WORDS, FROM WRFROM
025464  002470                          WRFROM                  ;STARTING FROM WRFROM
025466  000400                          256.                    ;256. WORDS
025470  125252                          125252                  ;FILL WITH 125252


                                        ;*NOW READ INTO BUFFER WILL BE FILLED WITH SAME DATA AS
                                        ;*WRITE FROM BUFFER SO THAT AFTER A WRITE COMPARISONS
                                        ;*CAN BE MADE TO DETERMINE THAT WRITE DID NOT CHANGE BUFFER

025472  004037  041374                  JSR     R0,@#CLAREA     ;CLEAR 256. WORDS, FROM REINTO
025476  003534                          REINTO                  ;STARTING FROM REINTO
025500  000400                          256.                    ;256. WORDS
025502  125252                          125252                  ;FILL WITH 125252


                                        ;*NOW WRITE DATA COMMAND WILL BE LOADED
```

```
6627
6628
6629  025504  004037  043476              JSR     R0,@#RUN        ;SETUP TO RUN FOR DATA COMMAND
6630  025510  000000                      0                       ;CYLINDER 0
6631  025512     000              .BYTE   0                       ;SECTOR 0
6632  025513     000              .BYTE   0                       ;TRACK 0
6633  025514  177400                      -256.                   ;WORD COUNT = 256.
6634  025516  002470                      WRFROM                  ;BUS ADDRESS
6635                                                              ;STARTING ADDRESS OF DATA
6636                                                              ;BUFFER = WRFROM
6637  025520  000000                      0                       ;DO NOT INHIBIT BUS ADDRESS INCREMENT
6638  025522  010000                      FMT22                   ;16 BITS PER WORD FORMAT
6639                                                              ;DO NOT INHIBIT ECC CORRECTION
6640                                                              ;DO NOT INHIBIT HEADER COMPARE
6641  025524  002442                      WRIDAT                  ;GET READY TO DO A WRIDAT
6642                                                    ;WRITE DATA WITH 60 IN RHCS1
6643
6644
6645                                      ;*NOW SAVE REGISTERS FOR COMPARISON AFTER WRITE DATA
6646
6647  025526  004037  041672              JSR     R0,@#SAVER      ;SAVE REGISTERS
6648  025532  002272                      RHWC                    ;RHWC IS THE FIRST REGISTER SAVED
6649  025534  004612                      SAVERE                  ;STARTING ADDRESS OF WHERE
6650                                                              ;THE REGISTERS ARE SAVED
6651  025536  000022                      18.                     ;NUMBER OF REGISTERS
6652                                                              ;SAVED = 18.
6653
6654  025540  004737  041604              JSR     PC,@#CHECKT     ;CHECK DVA,RDY,MOL,DPR,DRY,VV = 1
6655                                                              ;AND THAT NO STATUS BITS ARE STUCK = 1
6656  025544  104401  067033              TYPE    ,CPHALT         ;CANNOT CONTINUE TESTING IF ANY OF
6657                                                              ;THE FIRST SET OF BITS DON'T = 1
6658  025550  000000                      HALT                    ;STOP
6659
6660  025552  013777  004606  154506      MOV     @#RP4VEC,@RPVEC ;SET RP04 VECTOR ADDRESS
6661                                                              ;TO 'TIME1' IF P-CLOCK IS PRESENT
6662                                                              ;OR TO 'TIME2' IF P-CLOCK IS NOT PRESENT
6663                                                              ;'TIME' WILL ONLY SAVE
6664                                                              ;CURRENT CYLINDER ADDRESS
6665                                                              ;AND LOOK AHEAD REGISTERS
6666
6667
6668  025560  013746  002442              MOV     @#WRIDAT,-(SP)  ;GET READY TO MOVE COMMAND
6669  025564  052716  000101              BIS     #GO!IE,(SP)     ;GET READY TO SET 'GO' AND
6670                                                              ;ENABLE INTERRUPT
6671  025570  012677  154504              MOV     (SP)+,@RHCS1    ;GO WITH
6672                                                               ;60 IN RHCS1 FOR WRITE DATA
6673                                                              ;WITH INTERRUPT ENABLED
6674  025574  011100                      MOV     @R1,R0          ;SAVE RHCS1 DURING ABOVE OPERATION
6675  025576  011305                      MOV     @R3,R5          ;SAVE RHDS1 DURING ABOVE OPERATION
6676
6677                                      ;*ONE REVOLUTION=16670 MICROSEC, ONE SECTOR=760 MICROSEC
6678
6679
6680  025600  104413                      WAT                     ;WAIT FOR RDY BIT TO SET
6681  025602  002300                      RHCS1                   ;WAIT FOR RHCS1 REGISTER
6682  025604  000200                      RDY                     ;WAIT FOR RDY BIT IN RHCS1 REGISTER
```

```
6683  025606  001614                          908.                    ;ALLOW 9080 MICRO SECONDS
6684  025610  001507                          839.                    ;RDY MUST SET BETWEEN
6685                                                                  ;690 AND 17470 MICRO SECONDS
6686
6687                                          ;*COMPARE CONTENTS OF RHCS1 AND RHDS1 ALREADY SAVED IN
6688                                          ;*R0 AND R5 IMMEDIATELY AFTER GO
6689
6690  025612  013746  002442                  MOV     @#WR!DAT,-(SP)  ;SAVE COMMAND
6691  025616  052716  004101                  BIS     #IE!GO!DVA,(SP) ;INCLUDE IE!GO!DVA
6692  025622  011637  001124                  MOV     (SP),@#$GDDAT   ;SAVE FOR PRINTOUT
6693  025626  022600                          CMP     (SP)+,R0        ;DURING ABOVE OPERATION ONLY IE!GO!DVA
6694                                                                  ;AND COMMAND SHOULD BE SET
6695  025630  001405                          BEQ     64$             ;BRANCH IF GOOD
6696  025632  010037  001126                  MOV     R0,@#$BDDAT     ;BAD DATA
6697  025636  010137  004600                  MOV     R1,@#REGADR     ;FAILING REGISTER RHCS1
6698  025642  104021                          ERROR   21              ;DURING ABOVE OPERATION ONLY
6699                                                                  ;COMMAND AND IE!GO!DVA SHOULD BE SET
6700  025644  012746  010500          64$:    MOV     #MOL!DPR!VV,-(SP)       ;SAVE BITS SET DURING OPERATION IN RHDS1
6701  025650  011637  001124                  MOV     (SP),@#$GDDAT   ;SAVE FOR PRINTOUT
6702  025654  022605                          CMP     (SP)+,R5        ;DURING ABOVE OPERATION ONLY MOL!DPR!VV
6703                                                                  ;SHOULD BE SET
6704  025656  001405                          BEQ     66$             ;BRANCH IF GOOD
6705  025660  010537  001126                  MOV     R5,@#$BDDAT     ;BAD DATA
6706  025664  010337  004600                  MOV     R3,@#REGADR     ;FAILING REGISTER RHDS1
6707  025670  104063                          ERROR   63              ;DURING ABOVE OPERATION ONLY
6708                                                                  ;MOL!DPR!VV SHOULD BE SET
6709  025672                          66$:
6710
6711                                          ;*NOW CHANGE SAVED REGISTERS TO EXPECTED VALUE
6712
6713  025672  004037  041426                  JSR     R0,@#FILLRE     ;MOV 0 INTO SAVED RHWC
6714  025676  002272                          RHWC                    ;SAVED REGISTER TO CHANGE
6715  025700  000000                          0                       ;DATA
6716  025702  004037  041426                  JSR     R0,@#FILLRE     ;MOV WRFROM+<256.*2> INTO SAVED RHBA
6717  025706  002274                          RHBA                    ;SAVED REGISTER TO CHANGE
6718  025710  003470                          WRFROM+<256.*2>                 ;DATA
6719  025712  004037  041426                  JSR     R0,@#FILLRE     ;MOV 1 INTO SAVED RHDST
6720  025716  002304                          RHDST                   ;SAVED REGISTER TO CHANGE
6721  025720  000001                          1                       ;DATA
6722
6723                                          ;*NOW COMPARE REGISTERS BEFORE WRITE DATA WITH REGISTERS
6724                                          ;*AFTER COMMAND
6725
6726
6727  025722  004037  042512                  JSR     R0,@#COMREG     ;COMPARE SAVED REGISTERS WITH
6728                                                                  ;PRESENT VALUE
6729  025726  004612                          SAVERE                  ;GOOD DATA SAVED IN 'SAVERE'
6730  025730  002354                          WC                      ;TEST DATA STARTING FROM 'RHWC'
6731  025732  000022                          18.                     ;18. REGISTERS TO BE COMPARED
6732  025734  025740                          1$                      ;RETURN TO 1$ ON ERROR
6733  025736  025744                          2$                      ;RETURN TO 2$ ON NO ERROR
6734
6735  025740  104035                  1$:     ERROR   35              ;WRITE DATA COMMAND CAUSED
6736  025742  000207                          RTS     PC              ;IMPROPER REGISTER CHANGE
6737                                                                  ;GOOD DATA GIVES WHAT SHOULD
6738                                                                  ;BE
```

```
                                                        ;RECEIVED DATA GIVES WHAT WAS
                                                        ;THERE AFTER COMMAND

                                        ;*NOW WRITE FROM BUFFER WILL CHECKED FOR NO CHANGE

025744                          2$:

025744  004037  043542                  JSR     R0,@#COMPAR     ;COMPARE TWO BLOCKS OF MEMORY
025750  003534                          REINTO                  ;GOOD DATA STARTS FROM REINTO
025752  002470                          WRFROM                  ;TEST DATA STARTS FROM WRFROM
025754  000400                          256.                    ;256. WORDS TO BE COMPARED
025756  025762                          3$                      ;RETURN TO 3$ ON ERROR
025760  025766                          4$                      ;RETURN TO 4$ ON NO ERROR

025762  104036                  3$:     ERROR   36              ;WRITE DATA COMMAND CHANGED
025764  000207                          RTS     PC              ;WRITE FROM BUFFER

                                        ;*NOW A READ DATA COMMAND WILL BE GIVEN
                                        ;*FILL READ INTO BUFFER WITH 256 ZEROS

025766                          4$:

025766  004737  041524                  JSR     PC,@#CLDISK     ;SET R1-RHCS1, R2-RHCS2
                                                                ;R3-RHDS1, R4-RHER1
                                                                ;GIVE RH-11 INITIALIZE
                                                                ;SETUP UNIT NUMBER
025772  004037  041374                  JSR     R0,@#CLAREA     ;CLEAR 256. WORDS, FROM REINTO
025776  003534                          REINTO                  ;STARTING FROM REINTO
026000  000400                          256.                    ;256. WORDS
026002  000000                          0                       ;FILL WITH 0

                                        ;*FILL WRITE FROM BUFFER WITH 256 OF 125252

026004  004037  041374                  JSR     R0,@#CLAREA     ;CLEAR 256. WORDS, FROM WRFROM
026010  002470                          WRFROM                  ;STARTING FROM WRFROM
026012  000400                          256.                    ;256. WORDS
026014  125252                          125252                  ;FILL WITH 125252

                                        ;*NOW FILL COMMAND

026016  004037  043476                  JSR     R0,@#RUN        ;SETUP TO RUN FOR DATA COMMAND
026022  000000                          0                       ;CYLINDER 0
026024     000                  .BYTE   0                       ;SECTOR 0
026025     000                  .BYTE   0                       ;TRACK 0
026026  177400                          -256.                   ;WORD COUNT = 256.
026030  003534                          REINTO                  ;BUS ADDRESS
                                                                ;STARTING ADDRESS OF DATA
                                                                ;BUFFER = REINTO
026032  000000                          0                       ;DO NOT INHIBIT BUS ADDRESS INCREMENT
026034  014000                          EC1!FMT22                       ;16 BITS PER WORD FORMAT
                                                                ;INHIBIT ECC CORRECTION
                                                                ;DO NOT INHIBIT HEADER COMPARE
```

```
                                READAT                  ;GET READY TO DO A READAT
                                                        ;READ DATA WITH 70 IN RHCS1



                                ;*NOW SAVE REGISTERS FOR COMPARISON AFTER READ DATA COMMAND

026040  004037  041672          JSR     R0,@#SAVER      ;SAVE REGISTERS
026044  002272                  RHWC                    ;RHWC IS THE FIRST REGISTER SAVED
026046  004612                  SAVERE                  ;STARTING ADDRESS OF WHERE
                                                        ;THE REGISTERS ARE SAVED
026050  000022                  18.                     ;NUMBER OF REGISTERS
                                                        ;SAVED = 18.

026052  004737  041604          JSR     PC,@#CHECKT     ;CHECK DVA,RDY,MOL,DPR,DRY,VV = 1
                                                        ;AND THAT NO STATUS BITS ARE STUCK = 1
026056  104401  067033          TYPE    ,CPHALT         ;CANNOT CONTINUE TESTING IF ANY OF
                                                        ;THE FIRST SET OF BITS DON'T = 1
026062  000000                  HALT                    ;STOP

026064  013777  004606  154174  MOV     @#RP4VEC,@RPVEC ;SET RP04 VECTOR ADDRESS
                                                        ;TO 'TIME1' IF P-CLOCK IS PRESENT
                                                        ;OR TO 'TIME2' IF P-CLOCK IS NOT PRESENT
                                                        ;'TIME' WILL ONLY SAVE
                                                        ;CURRENT CYLINDER ADDRESS
                                                        ;AND LOOK AHEAD REGISTERS


026072  013746  002446          MOV     @#READAT,-(SP)  ;GET READY TO MOVE COMMAND
026076  052716  000101          BIS     #GO!IE,(SP)     ;GET READY TO SET 'GO' AND
                                                        ;ENABLE INTERRUPT
026102  012677  154172          MOV     (SP)+,@RHCS1    ;GO WITH
                                                         ;70 IN RHCS1 FOR READ DATA
                                                        ;WITH INTERRUPT ENABLED
026106  011100                  MOV     @R1,R0          ;SAVE RHCS1 DURING ABOVE OPERATION
026110  011305                  MOV     @R3,R5          ;SAVE RHDS1 DURING ABOVE OPERATION


026112  104413                  WAT                     ;WAIT FOR RDY BIT TO SET
026114  002300                  RHCS1                   ;WAIT FOR RHCS1 REGISTER
026116  000200                  RDY                     ;WAIT FOR RDY BIT IN RHCS1 REGISTER
026120  001614                  908.                    ;ALLOW 9080 MICRO SECONDS
026122  001507                  839.                    ;RDY MUST SET BETWEEN
                                                        ;690 AND 17470 MICRO SECONDS


                                ;*COMPARE CONTENTS OF RHCS1 AND RHDS1 ALREADY SAVED IN
                                ;*R0 AND R5 IMMEDIATELY AFTER GO

026124  013746  002446          MOV     @#READAT,-(SP)  ;SAVE COMMAND
026130  052716  004101          BIS     #IE!GO!DVA,(SP) ;INCLUDE IE!GO!DVA
026134  011637  001124          MOV     (SP),@#$GDDAT   ;SAVE FOR PRINTOUT
026140  022600                  CMP     (SP)+,R0        ;DURING ABOVE OPERATION ONLY IE!GO!DVA
                                                        ;AND COMMAND SHOULD BE SET
026142  001405                  BEQ     67$             ;BRANCH IF GOOD
026144  010037  001126          MOV     R0,@#$BDDAT     ;BAD DATA
026150  010137  004600          MOV     R1,@#REGADR     ;FAILING REGISTER RHCS1
026154  104021                  ERROR   21              ;DURING ABOVE OPERATION ONLY
```

```
                                                        ;COMMAND AND IE!GO!DVA SHOULD BE SET
026156  012746  010500          67$:  MOV   #MOL!DPR!VV,-(SP)     ;SAVE BITS SET DURING OPERATION IN RHDS1
026162  011637  001124                MOV   (SP),@#$GDDAT       ;SAVE FOR PRINTOUT
026166  022605                       CMP   (SP)+,R5            ;DURING ABOVE OPERATION ONLY MOL!DPR!VV
                                                        ;SHOULD BE SET
026170  001405                       BEQ   69$                 ;BRANCH IF GOOD
026172  010537  001126               MOV   R5,@#$BDDAT         ;BAD DATA
026176  010337  004600               MOV   R3,@#REGADR         ;FAILING REGISTER RHDS1
026202  104063                       ERROR 63                  ;DURING ABOVE OPERATION ONLY
                                                        ;MOL!DPR!VV SHOULD BE SET
026204                           69$:
                                ;*CHANGE SAVED REGISTERS TO EXPECTED VALUES
026204  004037  041426                JSR   R0,@#FILLRE         ;MOV 0 INTO SAVED RHWC
026210  002272                       RHWC                      ;SAVED REGISTER TO CHANGE
026212  000000                       0                         ;DATA
026214  004037  041426               JSR   R0,@#FILLRE         ;MOV REINTO+<256.*2> INTO SAVED RHBA
026220  002274                       RHBA                      ;SAVED REGISTER TO CHANGE
026222  004534                       REINTO+<256.*2>                 ;DATA
026224  004037  041426               JSR   R0,@#FILLRE         ;MOV 1 INTO SAVED RHDST
026230  002304                       RHDST                     ;SAVED REGISTER TO CHANGE
026232  000001                       1                         ;DATA
                                ;*COMPARE REGISTERS BEFORE READ DATA COMMAND
                                ;*WITH REGISTERS AFTER COMMAND
026234  004037  042512               JSR   R0,@#COMREG         ;COMPARE SAVED REGISTERS WITH
                                                        ;PRESENT VALUE
026240  004612                       SAVERE                    ;GOOD DATA SAVED IN 'SAVERE'
026242  002354                       WC                        ;TEST DATA STARTING FROM 'RHWC'
026244  000022                       18.                       ;18. REGISTERS TO BE COMPARED
026246  026252                       5$                        ;RETURN TO 5$ ON ERROR
026250  026256                       6$                        ;RETURN TO 6$ ON NO ERROR
026252  104033           5$:         ERROR 33                  ;READ DATA CAUSED IMPROPER
026254  000207                       RTS   PC                  ;REGISTER CHANGE
                                                        ;GOOD DATA GIVES WHAT SHOULD BE THE
                                                        ;RECEIVED DATA GIVES WHAT WAS THERE
                                                        ;AFTER COMMAND
                                ;*NOW READ INTO BUFFER IS CHECKED FOR GOOD READ
026256                           6$:
026256  004037  043542               JSR   R0,@#COMPAR         ;COMPARE TWO BLOCKS OF MEMORY
026262  002470                       WRFROM                    ;GOOD DATA STARTS FROM WRFROM
026264  003534                       REINTO                    ;TEST DATA STARTS FROM REINTO
026266  000400                       256.                      ;256. WORDS TO BE COMPARED
026270  026274                       7$                        ;RETURN TO 7$ ON ERROR
026272  026300                       10$                       ;RETURN TO 10$ ON NO ERROR
026274  104034           7$:         ERROR 34                  ;INCORRECT DATA AFTER
026276  000207                       RTS   PC                  ;WRITE DATA FOLLOWED BY A
```

```
 6907                                                    ;READ DATA
 6908  026300                        10$:
 6909
```

```
                                ;;*****************************************************************
                                ;*TEST 37       WRITE/READ DATA (052525)
                                ;*
                                ;*      THIS TEST GIVES A WRITE DATA COMMAND FOR CYLINDER 0
                                ;*      TRACK 0, SECTOR 0, KEYS 0, 200 WORDS OF 052525
                                ;*      THIS SECTOR IS FORMATED BY PREVIOUS TEST
                                ;*      THEN READ DATA COMMAND IS GIVEN FOR 256 WORDS IN
                                ;*      SAME CYLINDER, TRACK, SECTOR, KEYS
                                ;*      WRITE FROM BUFFER AND READ INTO BUFFER ARE FILLED WITH
                                ;*      200 OF 052525 AND 56 OF 377
                                ;*      THE WRITE DATA COMMAND IS LOADED EXCEPT GO AND IE
                                ;*      ALL REGISTERS ARE SAVED AND THEN GO IS TO WRITE DATI
                                ;*
                                ;*
                                ;*      THEN ALL REGISTERS ARE COMPARED TO CHECK FOR IMPROPER
                                ;*      CHANGE, THEN WRITE FROM BUFFER IS CHECKED FOR NO CHANGE
                                ;*
                                ;*      THEN READ INTO BUFFER IS FILLED WITH 200 IF ZEROS
                                ;*      AND 56 ALL ONES, WRITE FROM BUFFER IS FILLED WITH 200
                                ;*      WORDS OF 52525 AND 56 WORDS OF 0
                                ;*      THE COMMAND EXCEPT GO IS LOADED, ALL REGISTERS ARE SAVED
                                ;*      GO IS GIVEN
                                ;*
                                ;*
                                ;*      ALL REGISTER ARE CHECKED
                                ;*      READ DATA IS CHECKED

                                ;;*****************************************************************
                                ;;*****************************************************************
026300  000004          TST37:  SCOPE
026302  012706  001000          MOV     #STACK,SP       ;RESET STACK
026306  012737  000037  004604  MOV     #37,@#TSTNM     ;SAVE TEST NUMBER

026314  004737  041524          JSR     PC,@#CLDISK     ;SET R1-RHCS1, R2-RHCS2
                                                        ;R3-RHDS1, R4-RHER1
                                                        ;GIVE RH-11 INITIALIZE
                                                        ;SETUP UNIT NUMBER

                                ;*NOW FILL WRITE FROM BUFFER-200 OF 52525 AND 56 OF 377

026320  004037  041374          JSR     R0,@#CLAREA     ;CLEAR 200. WORDS, FROM WRFROM
026324  002470                  WRFROM                  ;STARTING FROM WRFROM
026326  000310                  200.                    ;200. WORDS
026330  052525                  52525                   ;FILL WITH 52525

026332  004037  041374          JSR     R0,@#CLAREA     ;CLEAR 56. WORDS, FROM WRFROM+<200.*2>
026336  003310                  WRFROM+<200.*2>                 ;STARTING FROM WRFROM+<200.*2>
026340  000070                  56.                     ;56. WORDS
026342  000377                  377                     ;FILL WITH 377


                                ;*NOW READ INTO BUFFER WILL BE FILLED WITH SAME DATA AS
                                ;*WRITE FROM BUFFER SO THAT AFTER A WRITE COMPARISONS
                                ;*CAN BE MADE TO DETERMINE THAT WRITE DID NOT CHANGE BUFFER

026344  004037  041374          JSR     R0,@#CLAREA     ;CLEAR 200. WORDS, FROM REINTO
026350  003534                  REINTO                  ;STARTING FROM REINTO
```

```
                                        200.                    ;200. WORDS
                                        52525                   ;FILL WITH 52525

                        JSR     R0,@#CLAREA     ;CLEAR 56. WORDS, FROM REINTO+<200.*2>
                        REINTO+<200.*2>                 ;STARTING FROM REINTO+<200.*2>
                                        56.                     ;56. WORDS
                                        377                     ;FILL WITH 377


                                        ;*NOW WRITE DATA COMMAND WILL BE LOADED


                        JSR     R0,@#RUN        ;SETUP TO RUN FOR DATA COMMAND
                                        0                       ;CYLINDER 0
                .BYTE   0                       ;SECTOR 0
                .BYTE   0                       ;TRACK 0
                                        -200.                   ;WORD COUNT = 200.
                                        WRFROM                  ;BUS ADDRESS
                                                                ;STARTING ADDRESS OF DATA
                                                                ;BUFFER = WRFROM
                                        0                       ;DO NOT INHIBIT BUS ADDRESS INCREMENT
                                        FMT22                   ;16 BITS PER WORD FORMAT
                                                                ;DO NOT INHIBIT ECC CORRECTION
                                                                ;DO NOT INHIBIT HEADER COMPARE
                                        WRIDAT                  ;GET READY TO DO A WRIDAT
                                                        ;WRITE DATA WITH 60 IN RHCS1


                                        ;*NOW SAVE REGISTER FOR COMPARISON AFTER WRITE DATA

                        JSR     R0,@#SAVER      ;SAVE REGISTERS
                        RHWC                    ;RHWC IS THE FIRST REGISTER SAVED
                        SAVERE                  ;STARTING ADDRESS OF WHERE
                                                ;THE REGISTERS ARE SAVED
                        18.                     ;NUMBER OF REGISTERS
                                                ;SAVED = 18.

                        JSR     PC,@#CHECKT     ;CHECK DVA,RDY,MOL,DPR,DRY,VV = 1
                                                ;AND THAT NO STATUS BITS ARE STUCK = 1
                        TYPE    ,CPHALT         ;CANNOT CONTINUE TESTING IF ANY OF
                                                ;THE FIRST SET OF BITS DON'T = 1
                        HALT                    ;STOP

                        MOV     @#RP4VEC,@RPVEC ;SET RP04 VECTOR ADDRESS
                                                ;TO 'TIME1' IF P-CLOCK IS PRESENT
                                                ;OR TO 'TIME2' IF P-CLOCK IS NOT PRESENT
                                                ;'TIME' WILL ONLY SAVE
                                                ;CURRENT CYLINDER ADDRESS
                                                ;AND LOOK AHEAD REGISTERS


                        MOV     @#WRIDAT,-(SP)  ;GET READY TO MOVE COMMAND
                        BIS     #GO!IE,(SP)     ;GET READY TO SET 'GO' AND
                                                ;ENABLE INTERRUPT
                        MOV     (SP)+,@RHCS1    ;GO WITH
                                                 ;60 IN RHCS1 FOR WRITE DATA
```

```
                                                        ;WITH INTERRUPT ENABLED
026460  011100                      MOV     @R1,R0      ;SAVE RHCS1 DURING ABOVE OPERATION
026462  011305                      MOV     @R3,R5      ;SAVE RHDS1 DURING ABOVE OPERATION

                                    ;*ONE REVOLUTION=16670 MICROSEC, ONE SECTOR=760 MICROSEC


026464  104413                      WAT                 ;WAIT FOR RDY BIT TO SET
026466  002300                      RHCS1               ;WAIT FOR RHCS1 REGISTER
026470  000200                      RDY                 ;WAIT FOR RDY BIT IN RHCS1 REGISTER
026472  001614                      908.                ;ALLOW 9080 MICRO SECONDS
026474  001507                      839.                ;RDY MUST SET BETWEEN
                                                        ;690 AND 17470 MICRO SECONDS


                                    ;*COMPARE CONTENTS OF RHCS1 AND RHDS1 ALREADY SAVED IN
                                    ;*R0 AND R5 IMMEDIATELY AFTER GO

026476  013746  002442              MOV     @#WRIDAT,-(SP)  ;SAVE COMMAND
026502  052716  004101              BIS     #IE!GO!DVA,(SP) ;INCLUDE IE!GO!DVA
026506  011637  001124              MOV     (SP),@#$GDDAT   ;SAVE FOR PRINTOUT
026512  022600                      CMP     (SP)+,R0        ;DURING ABOVE OPERATION ONLY IE!GO!DVA
                                                        ;AND COMMAND SHOULD BE SET
026514  001405                      BEQ     64$             ;BRANCH IF GOOD
026516  010037  001126              MOV     R0,@#$BDDAT     ;BAD DATA
026522  010137  004600              MOV     R1,@#REGADR     ;FAILING REGISTER RHCS1
026526  104021                      ERROR   21              ;DURING ABOVE OPERATION ONLY
                                                        ;COMMAND AND IE!GO!DVA SHOULD BE SET
026530  012746  010500      64$:    MOV     #MOL!DPR!VV,-(SP)       ;SAVE BITS SET DURING OPERATION IN RHDS1
026534  011637  001124              MOV     (SP),@#$GDDAT   ;SAVE FOR PRINTOUT
026540  022605                      CMP     (SP)+,R5        ;DURING ABOVE OPERATION ONLY MOL!DPR!VV
                                                        ;SHOULD BE SET
026542  001405                      BEQ     66$             ;BRANCH IF GOOD
026544  010537  001126              MOV     R5,@#$BDDAT     ;BAD DATA
026550  010337  004600              MOV     R3,@#REGADR     ;FAILING REGISTER RHDS1
026554  104063                      ERROR   63              ;DURING ABOVE OPERATION ONLY
                                                        ;MOL!DPR!VV SHOULD BE SET
026556                      66$:


                                    ;*NOW CHANGE SAVED REGISTERS TO EXPECTED VALUE

026556  004037  041426              JSR     R0,@#FILLRE     ;MOV 0 INTO SAVED RHWC
026562  002272                      RHWC                    ;SAVED REGISTER TO CHANGE
026564  000000                      0                       ;DATA
026566  004037  041426              JSR     R0,@#FILLRE     ;MOV WRFROM+<200.*2> INTO SAVED RHBA
026572  002274                      RHBA                    ;SAVED REGISTER TO CHANGE
026574  003310                      WRFROM+<200.*2>                 ;DATA
026576  004037  041426              JSR     R0,@#FILLRE     ;MOV 1 INTO SAVED RHDST
026602  002304                      RHDST                   ;SAVED REGISTER TO CHANGE
026604  000001                      1                       ;DATA


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


026606  004037  042512              JSR     R0,@#COMREG     ;COMPARE SAVED REGISTERS WITH
                                                        ;PRESENT VALUE
```

```
7078  026612  004612                      SAVERE                  ;GOOD DATA SAVED IN 'SAVERE'
7079  026614  002354                      WC                      ;TEST DATA STARTING FROM 'RHWC'
7080  026616  000022                      18.                     ;18. REGISTERS TO BE COMPARED
7081  026620  026624                      1$                      ;RETURN TO 1$ ON ERROR
7082  026622  026630                      2$                      ;RETURN TO 2$ ON NO ERROR
7083
7084  026624  104035              1$:     ERROR   35              ;WRITE DATA COMMAND CAUSED
7085  026626  000207                      RTS     PC              ;IMPROPER REGISTER CHANGE
7086                                                              ;GOOD DATA GIVES WHAT SHOULD
7087                                                              ;BE
7088                                                              ;RECEIVED DATA GIVES WHAT WAS
7089                                                              ;THERE AFTER COMMAND
7090
7091                                      ;*NOW WRITE FROM BUFFER WILL BE CHECKED FOR NO CHANGE
7092
7093  026630                      2$:
7094
7095  026630  004037  043542              JSR     R0,@#COMPAR     ;COMPARE TWO BLOCKS OF MEMORY
7096  026634  003534                      REINTO                  ;GOOD DATA STARTS FROM REINTO
7097  026636  002470                      WRFROM                  ;TEST DATA STARTS FROM WRFROM
7098  026640  000400                      256.                    ;256. WORDS TO BE COMPARED
7099  026642  026646                      3$                      ;RETURN TO 3$ ON ERROR
7100  026644  026652                      4$                      ;RETURN TO 4$ ON NO ERROR
7101
7102
7103  026646  104036              3$:     ERROR   36              ;WRITE DATA COMMAND CHANGED
7104  026650  000207                      RTS     PC              ;WRITE FROM BUFFER
7105
7106                                      ;*NOW A READ DATA COMMAND WILL BE GIVEN
7107
7108                                      ;*FILL READ INTO BUFFER WITH 200 ZEROS AND 56 OF ALL ONES
7109
7110  026652                      4$:
7111
7112  026652  004737  041524              JSR     PC,@#CLDISK     ;SET R1-RHCS1, R2-RHCS2
7113                                                              ;R3-RHDS1, R4-RHER1
7114                                                              ;GIVE RH-11 INITIALIZE
7115                                                              ;SETUP UNIT NUMBER
7116  026656  004037  041374              JSR     R0,@#CLAREA     ;CLEAR 200. WORDS, FROM REINTO
7117  026662  003534                      REINTO                  ;STARTING FROM REINTO
7118  026664  000310                      200.                    ;200. WORDS
7119  026666  000000                      0                       ;FILL WITH 0
7120
7121  026670  004037  041374              JSR     R0,@#CLAREA     ;CLEAR 56. WORDS, FROM REINTO+<200.*2>
7122  026674  004354                      REINTO+<200.*2>                 ;STARTING FROM REINTO+<200.*2>
7123  026676  000070                      56.                     ;56. WORDS
7124  026700  000377                      377                     ;FILL WITH 377
7125
7126
7127                                      ;*FILL WRITE FROM BUFFER WITH 200 OF 52525 AND 56 OF 0
7128
7129  026702  004037  041374              JSR     R0,@#CLAREA     ;CLEAR 200. WORDS, FROM WRFROM
7130  026706  002470                      WRFROM                  ;STARTING FROM WRFROM
7131  026710  000310                      200.                    ;200. WORDS
7132  026712  052525                      52525                   ;FILL WITH 52525
7133
```

```
7134  026714  004037  041374                  JSR     R0,@#CLAREA     ;CLEAR 56. WORDS, FROM WRFROM+<200.*2>
7135  026720  003310                          WRFROM+<200.*2>                 ;STARTING FROM WRFROM+<200.*2>
7136  026722  000070                          56.                     ;56. WORDS
7137  026724  000377                          377                     ;FILL WITH 377
7138
7139
7140                                          ;*NOW FILL COMMAND
7141
7142
7143  026726  004037  043476                  JSR     R0,@#RUN        ;SETUP TO RUN FOR DATA COMMAND
7144  026732  000000                          0                       ;CYLINDER 0
7145  026734     000                  .BYTE   0                       ;SECTOR 0
7146  026735     000                  .BYTE   0                       ;TRACK 0
7147  026736  177470                          -200.                   ;WORD COUNT = 200.
7148  026740  003534                          REINTO                  ;BUS ADDRESS
7149                                                                  ;STARTING ADDRESS OF DATA
7150                                                                  ;BUFFER = REINTO
7151  026742  000000                          0                       ;DO NOT INHIBIT BUS ADDRESS INCREMENT
7152  026744  014000                          ECI!FMT22                       ;16 BITS PER WORD FORMAT
7153                                                                  ;INHIBIT ECC CORRECTION
7154                                                                  ;DO NOT INHIBIT HEADER COMPARE
7155  026746  002446                          READAT                  ;GET READY TO DO A READAT
7156                                                        ;READ DATA WITH 70 IN RHCS1
7157
7158
7159                                          ;*NOW SAVE REGISTERS FOR COMPARISON AFTER READ DATA COMMAND
7160
7161  026750  004037  041672                  JSR     R0,@#SAVER      ;SAVE REGISTERS
7162  026754  002272                          RHWC                    ;RHWC IS THE FIRST REGISTER SAVED
7163  026756  004612                          SAVERE                  ;STARTING ADDRESS OF WHERE
7164                                                                  ;THE REGISTERS ARE SAVED
7165  026760  000022                          18.                     ;NUMBER OF REGISTERS
7166                                                                  ;SAVED = 18.
7167
7168  026762  004737  041604                  JSR     PC,@#CHECKT     ;CHECK DVA,RDY,MOL,DPR,DRY,VV = 1
7169                                                                  ;AND THAT NO STATUS BITS ARE STUCK = 1
7170  026766  104401  067033                  TYPE    ,CPHALT         ;CANNOT CONTINUE TESTING IF ANY OF
7171                                                                  ;THE FIRST SET OF BITS DON'T = 1
7172  026772  000000                          HALT                    ;STOP
7173
7174  026774  013777  004606  153264          MOV     @#RP4VEC,@RPVEC ;SET RP04 VECTOR ADDRESS
7175                                                                  ;TO 'TIME1' IF P-CLOCK IS PRESENT
7176                                                                  ;OR TO 'TIME2' IF P-CLOCK IS NOT PRESENT
7177                                                                  ;'TIME' WILL ONLY SAVE
7178                                                                  ;CURRENT CYLINDER ADDRESS
7179                                                                  ;AND LOOK AHEAD REGISTERS
7180
7181
7182  027002  013746  002446                  MOV     @#READAT,-(SP)  ;GET READY TO MOVE COMMAND
7183  027006  052716  000101                  BIS     #GO!IE,(SP)     ;GET READY TO SET 'GO' AND
7184                                                                  ;ENABLE INTERRUPT
7185  027012  012677  153262                  MOV     (SP)+,@RHCS1    ;GO WITH
7186                                                                   ;70 IN RHCS1 FOR READ DATA
7187                                                                  ;WITH INTERRUPT ENABLED
7188  027016  011100                          MOV     @R1,R0          ;SAVE RHCS1 DURING ABOVE OPERATION
7189  027020  011305                          MOV     @R3,R5          ;SAVE RHDS1 DURING ABOVE OPERATION
```

```
                                        WAT                     ;WAIT FOR RDY BIT TO SET
                                        RHCS1                   ;WAIT FOR RHCS1 REGISTER
                                        RDY                     ;WAIT FOR RDY BIT IN RHCS1 REGISTER
                                        908.                    ;ALLOW 9080 MICRO SECONDS
                                        839.                    ;RDY MUST SET BETWEEN
                                                                ;690 AND 17470 MICRO SECONDS

                                        ;*COMPARE CONTENTS OF RHCS1 AND RHDS1 ALREADY SAVED IN
                                        ;*R0 AND R5 IMMEDIATELY AFTER GO

                                        MOV     @#READAT,-(SP)  ;SAVE COMMAND
                                        BIS     #IE!GO!DVA,(SP) ;INCLUDE IE!GO!DVA
                                        MOV     (SP),@#$GDDAT   ;SAVE FOR PRINTOUT
                                        CMP     (SP)+,R0        ;DURING ABOVE OPERATION ONLY IE!GO!DVA
                                                                ;AND COMMAND SHOULD BE SET
                                        BEQ     67$             ;BRANCH IF GOOD
                                        MOV     R0,@#$BDDAT     ;BAD DATA
                                        MOV     R1,@#REGADR     ;FAILING REGISTER RHCS1
                                        ERROR   21              ;DURING ABOVE OPERATION ONLY
                                                                ;COMMAND AND IE!GO!DVA SHOULD BE SET
                                67$:    MOV     #MOL!DPR!VV,-(SP)       ;SAVE BITS SET DURING OPERATION IN RHDS1
                                        MOV     (SP),@#$GDDAT   ;SAVE FOR PRINTOUT
                                        CMP     (SP)+,R5        ;DURING ABOVE OPERATION ONLY MOL!DPR!VV
                                                                ;SHOULD BE SET
                                        BEQ     69$             ;BRANCH IF GOOD
                                        MOV     R5,@#$BDDAT     ;BAD DATA
                                        MOV     R3,@#REGADR     ;FAILING REGISTER RHDS1
                                        ERROR   63              ;DURING ABOVE OPERATION ONLY
                                                                ;MOL!DPR!VV SHOULD BE SET
                                69$:

                                        ;*CHANGE SAVED REGISTERS TO EXPECTED VALUES

                                        JSR     R0,@#FILLRE     ;MOV 0 INTO SAVED RHWC
                                        RHWC                    ;SAVED REGISTER TO CHANGE
                                        0                       ;DATA
                                        JSR     R0,@#FILLRE     ;MOV REINTO+<200.*2> INTO SAVED RHBA
                                        RHBA                    ;SAVED REGISTER TO CHANGE
                                        REINTO+<200.*2>                 ;DATA
                                        JSR     R0,@#FILLRE     ;MOV 1 INTO SAVED RHDST
                                        RHDST                   ;SAVED REGISTER TO CHANGE
                                        1                       ;DATA

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


                                        JSR     R0,@#COMREG     ;COMPARE SAVED REGISTERS WITH
                                                                ;PRESENT VALUE
                                        SAVERE                  ;GOOD DATA SAVED IN 'SAVERE'
                                        WC                      ;TEST DATA STARTING FROM 'RHWC'
                                        18.                     ;18. REGISTERS TO BE COMPARED
                                        5$                      ;RETURN TO 5$ ON ERROR
                                        6$                      ;RETURN TO 6$ ON NO ERROR
```

```
7246
7247  027162  104033                5$:     ERROR   33              ;READ DATA CAUSED IMPROPER
7248  027164  000207                        RTS     PC              ;REGISTER CHANGE
7249                                                                ;GOOD DATA GIVES WHAT SHOULD BE THE
7250                                                                ;RECEIVED DATA GIVES WHAT WAS THERE
7251                                                                ;AFTER COMMAND
7252
7253                                        ;*NOW READ INTO BUFFER IS CHECKED FOR GOOD READ
7254
7255  027166                        6$:
7256
7257  027166  004037  043542               JSR     R0,@#COMPAR     ;COMPARE TWO BLOCKS OF MEMORY
7258  027172  002470                        WRFROM                  ;GOOD DATA STARTS FROM WRFROM
7259  027174  003534                        REINTO                  ;TEST DATA STARTS FROM REINTO
7260  027176  000400                        256.                    ;256. WORDS TO BE COMPARED
7261  027200  027204                        7$                      ;RETURN TO 7$ ON ERROR
7262  027202  027210                        10$                     ;RETURN TO 10$ ON NO ERROR
7263
7264
7265  027204  104034                7$:     ERROR   34              ;INCORRECT DATA AFTER
7266  027206  000207                        RTS     PC              ;WRITE DATA FOLLOWED BY A
7267                                                                ;READ DATA
7268  027210                        10$:
7269
7270
7271
7272
```

```
                                ;;*****************************************************************
                                ;*TEST 40       WRITE/READ DATA USING UNIBUS B
                                ;
                                ;*      THIS TEST USES UNIBUS B IF CONNECTED TO THE RH
                                ;*      IF UNIBUS B IS NOT CONNECTED THEN THIS TEST IS NOT PERFORMED
                                ;*      THIS TEST GIVES A WRITE DATA COMMAND FOR CYLINDER 0
                                ;*      TRACK 0, SECTOR 0, KEYS 0, 200 WORDS OF 052525
                                ;*      THIS SECTOR IS FORMATED BY PREVIOUS TEST
                                ;*      THEN READ DATA COMMAND IS GIVEN FOR 256 WORDS IN
                                ;*      SAME CYLINDER, TRACK, SECTOR, KEYS
                                ;*      THESE COMMANDS USE UNIBUS B FOR DATA
                                ;*      WRITE FROM BUFFER AND READ INTO BUFFER ARE FILLED WITH
                                ;*      200 OF 052525 AND 56 OF 377
                                ;*      THE WRITE DATA COMMAND IS LOADED EXCEPT GO AND IE
                                ;*      ALL REGISTERS ARE SAVED AND THEN GO IS GIVEN TO WRITE DATA
                                ;*
                                ;*
                                ;*      THEN ALL REGISTERS ARE COMPARED TO CHECK FOR IMPROPER
                                ;*      CHANGE, THEN WRITE FROM BUFFER IS CHECKED FOR NO CHANGE
                                ;*
                                ;*      THEN READ INTO BUFFER IS FILLED WITH 200 IF ZEROS
                                ;*      AND 56 ALL ONES, WRITE FROM BUFFER IS FILLED WITH 200
                                ;*      WORDS OF 52525 AND 56 WORDS OF 0
                                ;*      THE COMMAND EXCEPT GO IS LOADED, ALL REGISTERS ARE SAVED
                                ;*      GO IS GIVEN
                                ;*
                                ;*
                                ;*      ALL REGISTER ARE CHECKED
                                ;*      READ DATA IS CHECKED
                                ;;*****************************************************************
027210  000004                TST40:  SCOPE
027212  012737  000001  001212        MOV     #1,$TIMES       ;;DO 1 ITERATION
027220  012706  001000                MOV     #STACK,SP       ;RESET STACK
027224  012737  000040  004604        MOV     #40,@#TSTNM     ;SAVE TEST NUMBER
                                
027232  004737  041524                JSR     PC,@#CLDISK     ;SET R1-RHCS1, R2-RHCS2
                                                            ;R3-RHDS1, R4-RHER1
                                                            ;GIVE RH-11 INITIALIZE
                                                            ;SETUP UNIT NUMBER
                                
                                        ;*CHECK TO SEE IF THE PROGRAM IS RUNNING WITH AN RH70
                                
027236  005737  004750                TST     @#RH70          ;TEST FOR RH70 CONTROLLER
027242  001402                        BEQ     30$             ;IF FLAG = 1, THIS TEST IS SKIPPED
027244  000137  030360                JMP     TST41   ;       JUMP TO NEXT TEST ------------------)
027250                          30$:                          ;IF FLAG = 1, DO THIS TEST
                                
027250  005037  004732                CLR     @#UBUSB         ;CLEAR UNIBUS INDICATOR
                                
                                        ;*NOW FILL WRITE FROM BUFFER-200 OF 52525 AND 56 OF 377
                                
027254  004037  041374                JSR     R0,@#CLAREA     ;CLEAR 200. WORDS, FROM WRFROM
027260  002470                        WRFROM                  ;STARTING FROM WRFROM
027262  000310                        200.                    ;200. WORDS
027264  052525                        52525                   ;FILL WITH 52525
```

```
7329
7330  027266  004037  041374              JSR     R0,@#CLAREA     ;CLEAR 56. WORDS, FROM WRFROM+<200.*2>
7331  027272  003310                      WRFROM+<200.*2>                 ;STARTING FROM WRFROM+<200.*2>
7332  027274  000070                      56.                     ;56. WORDS
7333  027276  000377                      377                     ;FILL WITH 377
7334
7335
7336                                      ;*NOW READ INTO BUFFER WILL BE FILLED WITH SAME DATA AS
7337                                      ;*WRITE FROM BUFFER SO THAT AFTER A WRITE COMPARISONS
7338                                      ;*CAN BE MADE TO DETERMINE THAT WRITE DID NOT CHANGE BUFFER
7339
7340  027300  004037  041374              JSR     R0,@#CLAREA     ;CLEAR 200. WORDS, FROM REINTO
7341  027304  003534                      REINTO                  ;STARTING FROM REINTO
7342  027306  000310                      200.                    ;200. WORDS
7343  027310  052525                      52525                   ;FILL WITH 52525
7344
7345  027312  004037  041374              JSR     R0,@#CLAREA     ;CLEAR 56. WORDS, FROM REINTO+<200.*2>
7346  027316  004354                      REINTO+<200.*2>                 ;STARTING FROM REINTO+<200.*2>
7347  027320  000070                      56.                     ;56. WORDS
7348  027322  000377                      377                     ;FILL WITH 377
7349
7350
7351                                      ;*NOW WRITE DATA COMMAND WILL BE LOADED
7352
7353
7354  027324  004037  043476              JSR     R0,@#RUN        ;SETUP TO RUN FOR DATA COMMAND
7355  027330  000000                      0                       ;CYLINDER 0
7356  027332     000              .BYTE   0                       ;SECTOR 0
7357  027333     000              .BYTE   0                       ;TRACK 0
7358  027334  177470                      -200.                   ;WORD COUNT = 200.
7359  027336  002470                      WRFROM                  ;BUS ADDRESS
7360                                                              ;STARTING ADDRESS OF DATA
7361                                                              ;BUFFER = WRFROM
7362  027340  000000                      0                       ;DO NOT INHIBIT BUS ADDRESS INCREMENT
7363  027342  010000                      FMT22                   ;16 BITS PER WORD FORMAT
7364                                                              ;DO NOT INHIBIT ECC CORRECTION
7365                                                              ;DO NOT INHIBIT HEADER COMPARE
7366  027344  002442                      WRIDAT                  ;GET READY TO DO A WRIDAT
7367                                                  ;WRITE DATA WITH 60 IN RHCS1
7368
7369  027346  052777  002000  152724      BIS     #PSEL,@RHCS1    ;SET PORT B
7370                                                              ;THAT IS UNIBUS B
7371
7372                                      ;*NOW SAVE REGISTER FOR COMPARISON AFTER WRITE DATA
7373
7374  027354  004037  041672              JSR     R0,@#SAVER      ;SAVE REGISTERS
7375  027360  002272                      RHWC                    ;RHWC IS THE FIRST REGISTER SAVED
7376  027362  004612                      SAVERE                  ;STARTING ADDRESS OF WHERE
7377                                                              ;THE REGISTERS ARE SAVED
7378  027364  000022                      18.                     ;NUMBER OF REGISTERS
7379                                                              ;SAVED = 18.
7380
7381  027366  004737  041604              JSR     PC,@#CHECKT     ;CHECK DVA,RDY,MOL,DPR,DRY,VV = 1
7382                                                              ;AND THAT NO STATUS BITS ARE STUCK = 1
7383  027372  104401  067033              TYPE    ,CPHALT         ;CANNOT CONTINUE TESTING IF ANY OF
7384                                                              ;THE FIRST SET OF BITS DON'T = 1
```

```
 7385  027376  000000                          HALT                    ;STOP
 7386
 7387  027400  013777  004606  152660          MOV     @#RP4VEC,@RPVEC ;SET RP04 VECTOR ADDRESS
 7388                                                                  ;TO 'TIME1' IF P-CLOCK IS PRESENT
 7389                                                                  ;OR TO 'TIME2' IF P-CLOCK IS NOT PRESENT
 7390                                                                  ;'TIME' WILL ONLY SAVE
 7391                                                                  ;CURRENT CYLINDER ADDRESS
 7392                                                                  ;AND LOOK AHEAD REGISTERS
 7393
 7394  027406  013746  002442                  MOV     @#WRIDAT,-(SP)  ;GET READY TO MOVE COMMAND
 7395  027412  052716  002101                  BIS     #GO!IE!PSEL,(SP)        ;GET READY TO SET 'GO' AND
 7396                                                                  ;ENABLE INTERRUPT
 7397  027416  012677  152656                  MOV     (SP)+,@RHCS1    ;GO WITH
 7398                                                                   ;60 IN RHCS1 FOR WRITE DATA
 7399                                                                  ;WITH INTERRUPT ENABLED
 7400  027422  011100                          MOV     @R1,R0          ;SAVE RHCS1 DURING ABOVE OPERATION
 7401  027424  011305                          MOV     @R3,R5          ;SAVE RHDS1 DURING ABOVE OPERATION
 7402
 7403                                          ;*ONE REVOLUTION=16670 MICROSEC, ONE SECTOR=760 MICROSEC
 7404
 7405
 7406  027426  104413                          WAT                     ;WAIT FOR RDY BIT TO SET
 7407  027430  002300                          RHCS1                   ;WAIT FOR RHCS1 REGISTER
 7408  027432  000200                          RDY                     ;WAIT FOR RDY BIT IN RHCS1 REGISTER
 7409  027434  001614                          908.                    ;ALLOW 9080 MICRO SECONDS
 7410  027436  001507                          839.                    ;RDY MUST SET BETWEEN
 7411                                                                  ;690 AND 17470 MICRO SECONDS
 7412
 7413                                          ;*COMPARE CONTENTS OF RHCS1 AND RHDS1 ALREADY SAVED IN
 7414                                          ;*R0 AND R5 IMMEDIATELY AFTER GO
 7415
 7416  027440  013746  002442                  MOV     @#WRIDAT,-(SP)  ;SAVE COMMAND
 7417  027444  052716  006101                  BIS     #IE!GO!DVA!PSEL,(SP)    ;INCLUDE IE!GO!DVA!PSEL
 7418  027450  011637  001124                  MOV     (SP),@#$GDDAT   ;SAVE FOR PRINTOUT
 7419  027454  022600                          CMP     (SP)+,R0        ;DURING ABOVE OPERATION ONLY IE!GO!DVA!PSEL
 7420                                                                  ;AND COMMAND SHOULD BE SET
 7421  027456  001405                          BEQ     64$             ;BRANCH IF GOOD
 7422  027460  010037  001126                  MOV     R0,@#$BDDAT     ;BAD DATA
 7423  027464  010137  004600                  MOV     R1,@#REGADR     ;FAILING REGISTER RHCS1
 7424  027470  104021                          ERROR   21              ;DURING ABOVE OPERATION ONLY
 7425                                                                  ;COMMAND AND IE!GO!DVA!PSEL SHOULD BE SET
 7426  027472  012746  010500          64$:    MOV     #MOL!DPR!VV,-(SP)       ;SAVE BITS SET DURING OPERATION IN RHDS1
 7427  027476  011637  001124                  MOV     (SP),@#$GDDAT   ;SAVE FOR PRINTOUT
 7428  027502  022605                          CMP     (SP)+,R5        ;DURING ABOVE OPERATION ONLY MOL!DPR!VV
 7429                                                                  ;SHOULD BE SET
 7430  027504  001405                          BEQ     66$             ;BRANCH IF GOOD
 7431  027506  010537  001126                  MOV     R5,@#$BDDAT     ;BAD DATA
 7432  027512  010337  004600                  MOV     R3,@#REGADR     ;FAILING REGISTER RHDS1
 7433  027516  104063                          ERROR   63              ;DURING ABOVE OPERATION ONLY
 7434                                                                  ;MOL!DPR!VV SHOULD BE SET
 7435  027520                          66$:
 7436
 7437
 7438                                          ;*CHECK IF NEM NON EXISTANT MEMORY IS SET
 7439                                          ;*IF SET IT MEANS UNIBUS B IS NOT CONNECTED
 7440                                          ;*SO THIS TEST IS NOT PERFORMED
```

```
                                                ;TEST NEM
                                                ;BRANCH IF UNIBUS B THERE
                                                ;UNIBUS B NOT THERE
                                                ;;TYPE ASCIZ STRING
                                                ;;GET OVER THE ASCIZ
                        ;;68$:  .ASCIZ  <15><12>/THE RH DOES NOT HAVE UNIBUS B CONNECTED/
                        67$:
                                                ;JUMP TO NEXT TEST - NO UNIBUS B
                        11$:
                                                ;;TYPE ASCIZ STRING
                                                ;;GET OVER THE ASCIZ
                        ;;70$:  .ASCIZ  <15><12>/THE RH DOES HAVE UNI BUS B CONNECTED/
                        69$:

                                ;*NOW CHANGE SAVED REGISTERS TO EXPECTED VALUE

                                                ;MOV 0 INTO SAVED RHWC
                                                ;SAVED REGISTER TO CHANGE
                                                ;DATA
                                                ;MOV WRFROM+<200.*2> INTO SAVED RHBA
                                                ;SAVED REGISTER TO CHANGE
                                                ;DATA
                                                ;MOV 1 INTO SAVED RHDST
                                                ;SAVED REGISTER TO CHANGE
                                                ;DATA

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


                                                ;COMPARE SAVED REGISTERS WITH
                                                ;PRESENT VALUE
                                                ;GOOD DATA SAVED IN 'SAVERE'
                                                ;TEST DATA STARTING FROM 'RHWC'
                                                ;18. REGISTERS TO BE COMPARED
                                                ;RETURN TO 1$ ON ERROR
                                                ;RETURN TO 2$ ON NO ERROR

                        1$:                     ;WHILE USING UNIBUS B
                                                ;WRITE DATA COMMAND CAUSED
                                                ;IMPROPER REGISTER CHANGE
                                                ;GOOD DATA GIVES WHAT SHOULD
                                                ;BE
                                                ;RECEIVED DATA GIVES WHAT WAS
                                                ;THERE AFTER COMMAND

                                ;*NOW WRITE FROM BUFFER WILL BE CHECKED FOR NO CHANGE

                        2$:

                                                ;COMPARE TWO BLOCKS OF MEMORY
```

```
7497  027776  003534                        REINTO                  ;GOOD DATA STARTS FROM REINTO
7498  030000  002470                        WRFROM                  ;TEST DATA STARTS FROM WRFROM
7499  030002  000400                        256.                    ;256. WORDS TO BE COMPARED
7500  030004  030010                        3$                      ;RETURN TO 3$ ON ERROR
7501  030006  030014                        4$                      ;RETURN TO 4$ ON NO ERROR
7502
7503
7504  030010  104075              3$:       ERROR   75              ;WHILE USING UNIBUS B
7505                                                                ;WRITE DATA COMMAND CHANGED
7506  030012  000207                        RTS     PC              ;WRITE FROM BUFFER
7507
7508                                        ;*NOW A READ DATA COMMAND WILL BE GIVEN
7509                                        ;*FILL READ INTO BUFFER WITH 200 ZEROS AND 56 OF ALL ONES
7510
7511  030014                        4$:
7512
7513  030014  004737  041524              JSR     PC,@#CLDISK     ;SET R1-RHCS1, R2-RHCS2
7514                                                                ;R3-RHDS1, R4-RHER1
7515                                                                ;GIVE RH-11 INITIALIZE
7516                                                                ;SETUP UNIT NUMBER
7517  030020  004037  041374              JSR     R0,@#CLAREA     ;CLEAR 200. WORDS, FROM REINTO
7518  030024  003534                      REINTO                  ;STARTING FROM REINTO
7519  030026  000310                      200.                    ;200. WORDS
7520  030030  000000                      0                       ;FILL WITH 0
7521
7522  030032  004037  041374              JSR     R0,@#CLAREA     ;CLEAR 56. WORDS, FROM REINTO+<200.*2>
7523  030036  004354                      REINTO+<200.*2>                 ;STARTING FROM REINTO+<200.*2>
7524  030040  000070                      56.                     ;56. WORDS
7525  030042  000377                      377                     ;FILL WITH 377
7526
7527
7528                                        ;*FILL WRITE FROM BUFFER WITH 200 OF 52525 AND 56 OF 0
7529
7530  030044  004037  041374              JSR     R0,@#CLAREA     ;CLEAR 200. WORDS, FROM WRFROM
7531  030050  002470                      WRFROM                  ;STARTING FROM WRFROM
7532  030052  000310                      200.                    ;200. WORDS
7533  030054  052525                      52525                   ;FILL WITH 52525
7534
7535  030056  004037  041374              JSR     R0,@#CLAREA     ;CLEAR 56. WORDS, FROM WRFROM+<200.*2>
7536  030062  003310                      WRFROM+<200.*2>                 ;STARTING FROM WRFROM+<200.*2>
7537  030064  000070                      56.                     ;56. WORDS
7538  030066  000377                      377                     ;FILL WITH 377
7539
7540
7541                                        ;*NOW FILL COMMAND
7542
7543
7544  030070  004037  043476              JSR     R0,@#RUN        ;SETUP TO RUN FOR DATA COMMAND
7545  030074  000000                      0                       ;CYLINDER 0
7546  030076     000              .BYTE   0                       ;SECTOR 0
7547  030077     000              .BYTE   0                       ;TRACK 0
7548  030100  177470                      -200.                   ;WORD COUNT = 200.
7549  030102  003534                      REINTO                  ;BUS ADDRESS
7550                                                                ;STARTING ADDRESS OF DATA
7551                                                                ;BUFFER = REINTO
7552  030104  000000                      0                       ;DO NOT INHIBIT BUS ADDRESS INCREMENT
```

```
7553  030106  014000                          ECI!FMT22                       ;16 BITS PER WORD FORMAT
7554                                                                        ;INHIBIT ECC CORRECTION
7555                                                                        ;DO NOT INHIBIT HEADER COMPARE
7556  030110  002446                          READAT                          ;GET READY TO DO A READAT
7557                                                                  ;READ DATA WITH 70 IN RHCS1
7558
7559  030112  052777  002000  152160          BIS     #PSEL,@RHCS1    ;SET PORT B
7560                                                                        ;THAT IS UNIBUS B
7561
7562                                          ;*NOW SAVE REGISTERS FOR COMPARISON AFTER READ DATA COMMAND
7563
7564  030120  004037  041672                  JSR     R0,@#SAVER      ;SAVE REGISTERS
7565  030124  002272                          RHWC                            ;RHWC IS THE FIRST REGISTER SAVED
7566  030126  004612                          SAVERE                          ;STARTING ADDRESS OF WHERE
7567                                                                        ;THE REGISTERS ARE SAVED
7568  030130  000022                          18.                             ;NUMBER OF REGISTERS
7569                                                                        ;SAVED = 18.
7570
7571  030132  004737  041604                  JSR     PC,@#CHECKT     ;CHECK DVA,RDY,MOL,DPR,DRY,VV = 1
7572                                                                        ;AND THAT NO STATUS BITS ARE STUCK = 1
7573  030136  104401  067033                  TYPE    ,CPHALT                 ;CANNOT CONTINUE TESTING IF ANY OF
7574                                                                        ;THE FIRST SET OF BITS DON'T = 1
7575  030142  000000                          HALT                            ;STOP
7576
7577  030144  013777  004606  152114          MOV     @#RP4VEC,@RPVEC ;SET RP04 VECTOR ADDRESS
7578                                                                        ;TO 'TIME1' IF P-CLOCK IS PRESENT
7579                                                                        ;OR TO 'TIME2' IF P-CLOCK IS NOT PRESENT
7580                                                                        ;'TIME' WILL ONLY SAVE
7581                                                                        ;CURRENT CYLINDER ADDRESS
7582                                                                        ;AND LOOK AHEAD REGISTERS
7583
7584  030152  013746  002446                  MOV     @#READAT,-(SP)  ;GET READY TO MOVE COMMAND
7585  030156  052716  002101                  BIS     #GO!IE!PSEL,(SP)        ;GET READY TO SET 'GO' AND
7586                                                                        ;ENABLE INTERRUPT
7587  030162  012677  152112                  MOV     (SP)+,@RHCS1    ;GO WITH
7588                                                                         ;70 IN RHCS1 FOR READ DATA
7589                                                                        ;WITH INTERRUPT ENABLED
7590  030166  011100                          MOV     @R1,R0          ;SAVE RHCS1 DURING ABOVE OPERATION
7591  030170  011305                          MOV     @R3,R5          ;SAVE RHDS1 DURING ABOVE OPERATION
7592
7593
7594  030172  104413                          WAT                             ;WAIT FOR RDY BIT TO SET
7595  030174  002300                          RHCS1                           ;WAIT FOR RHCS1 REGISTER
7596  030176  000200                          RDY                             ;WAIT FOR RDY BIT IN RHCS1 REGISTER
7597  030200  001614                          908.                            ;ALLOW 9080 MICRO SECONDS
7598  030202  001507                          839.                            ;RDY MUST SET BETWEEN
7599                                                                        ;690 AND 17470 MICRO SECONDS
7600
7601                                          ;*COMPARE CONTENTS OF RHCS1 AND RHDS1 ALREADY SAVED IN
7602                                          ;*R0 AND R5 IMMEDIATELY AFTER GO
7603
7604  030204  013746  002446                  MOV     @#READAT,-(SP)  ;SAVE COMMAND
7605  030210  052716  006101                  BIS     #IE!GO!DVA!PSEL,(SP)    ;INCLUDE IE!GO!DVA!PSEL
7606  030214  011637  001124                  MOV     (SP),@#$GDDAT   ;SAVE FOR PRINTOUT
7607  030220  022600                          CMP     (SP)+,R0        ;DURING ABOVE OPERATION ONLY IE!GO!DVA!PSEL
7608                                                                        ;AND COMMAND SHOULD BE SET
```

```
                                                BEQ     71$                 ;BRANCH IF GOOD
                                                MOV     R0,@#$BDDAT         ;BAD DATA
                                                MOV     R1,@#REGADR         ;FAILING REGISTER RHCS1
                                                ERROR   21                  ;DURING ABOVE OPERATION ONLY
                                                                            ;COMMAND AND IE!GO!DVA!PSEL SHOULD BE SET
                                        71$:    MOV     #MOL!DPR!VV,-(SP)       ;SAVE BITS SET DURING OPERATION IN RHDS1
                                                MOV     (SP),@#$GDDAT       ;SAVE FOR PRINTOUT
                                                CMP     (SP)+,R5            ;DURING ABOVE OPERATION ONLY MOL!DPR!VV
                                                                            ;SHOULD BE SET
                                                BEQ     73$                 ;BRANCH IF GOOD
                                                MOV     R5,@#$BDDAT         ;BAD DATA
                                                MOV     R3,@#REGADR         ;FAILING REGISTER RHDS1
                                                ERROR   63                  ;DURING ABOVE OPERATION ONLY
                                                                            ;MOL!DPR!VV SHOULD BE SET
                                        73$:

                                                ;*CHANGE SAVED REGISTERS TO EXPECTED VALUES

                                                JSR     R0,@#FILLRE         ;MOV 0 INTO SAVED RHWC
                                                RHWC                        ;SAVED REGISTER TO CHANGE
                                                0                           ;DATA
                                                JSR     R0,@#FILLRE         ;MOV REINTO+<200.*2> INTO SAVED RHBA
                                                RHBA                        ;SAVED REGISTER TO CHANGE
                                                REINTO+<200.*2>                 ;DATA
                                                JSR     R0,@#FILLRE         ;MOV 1 INTO SAVED RHDST
                                                RHDST                       ;SAVED REGISTER TO CHANGE
                                                1                           ;DATA

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


                                                JSR     R0,@#COMREG         ;COMPARE SAVED REGISTERS WITH
                                                                            ;PRESENT VALUE
                                                SAVERE                      ;GOOD DATA SAVED IN 'SAVERE'
                                                WC                          ;TEST DATA STARTING FROM 'RHWC'
                                                18.                         ;18. REGISTERS TO BE COMPARED
                                                5$                          ;RETURN TO 5$ ON ERROR
                                                6$                          ;RETURN TO 6$ ON NO ERROR

                                        5$:     ERROR   72                  ;WHILE USING UNIBUS B
                                                                            ;READ DATA CAUSED IMPROPER
                                                RTS     PC                  ;REGISTER CHANGE
                                                                            ;GOOD DATA GIVES WHAT SHOULD BE THE
                                                                            ;RECEIVED DATA GIVES WHAT WAS THERE
                                                                            ;AFTER COMMAND


                                                ;*NOW READ INTO BUFFER IS CHECKED FOR GOOD READ

                                        6$:

                                                JSR     R0,@#COMPAR         ;COMPARE TWO BLOCKS OF MEMORY
                                                WRFROM                      ;GOOD DATA STARTS FROM WRFROM
                                                REINTO                      ;TEST DATA STARTS FROM REINTO
                                                256.                        ;256. WORDS TO BE COMPARED
                                                7$                          ;RETURN TO 7$ ON ERROR
```

```
 7665  030352  030360                      10$                     ;RETURN TO 10$ ON NO ERROR
 7666
 7667
 7668  030354  104073              7$:     ERROR   73              ;WHILE USING UNIBUS B
 7669                                                              ;INCORRECT DATA AFTER
 7670  030356  000207                      RTS     PC              ;WRITE DATA FOLLOWED BY A
 7671                                                              ;READ DATA
 7672  030360                      10$:
 7673
 7674
 7675
 7676
 7677
 7678
```

```
7679
7680                            ;;*****************************************************************
7681                            ;*TEST 41       IMPLIED SEARCH
7682
7683                            ;*      ONLY NEW ADDITION !N THIS TEST IS AN IMPLIED SEARCH
7684                            ;*      A WRITE HEADER AND DATA IS GIVEN FOR MORE THAN ONE SECTOR
7685                            ;*      OF WORDS
7686                            ;*      WRITE HEADER AND DATA CYLINDER 0, FORMAT 16 BITS PER WORD
7687                            ;*      TRACK 0, SECTOR 0, KEYS=0, NUMBER OF WORDS
7688                            ;*      266 (4 HEADER 256 DATA=0 4 HEADER 2 DATA=1
7689                            ;*      THEN READ HEADER AND DATA FOR ABOVE SECTOR 1 ONLY
7690                            ;*      WRITE FROM BUFFER AND READ INTO BUFFER ARE FILLED WITH
7691                            ;*      10000,0,0,0 AND 256 OF 0, 10000,1,0,0, AND 2 OF 1
7692                            ;*      THE WRITE COMMAND IS THEN LOADED INTO THE REGISTERS EXCEPT
7693                            ;*      THE GO BIT, AND ALL THE REGISTERS ARE SAVED
7694                            ;*      THEN GO IS GIVEN FOR WRITE HEADER AND DATA
7695                            ;*
7696                            ;*      THEN ALL REGISTERS ARE COMPARED TO CHECK FOR IMPROPER CHANGE
7697                            ;*      THEN WRITE FROM BUFFER IS CHECKED TO SEE THAT NOTHING CHANGED
7698                            ;*
7699                            ;*
7700                            ;*      NOW FOR THE READ COMMAND READ INTO BUFFER IS FILLED
7701                            ;*      WITH 377, WRITE FROM BUFFER IS FILLED WITH 10000,1,0,0,1,1
7702                            ;*      AND 254 OF ZEROS COMMAND IS LOADED INTO REGISTERS EXCEPT
7703                            ;*      GO AND IE THEN ALL REGISTERS ARE SAVED
7704                            ;*      GO IS GIVEN FOR THE READ COMMAND, 256 WORDS
7705                            ;*
7706                            ;*      ALL REGISTERS ARE CHECKED FOR IMPROPER CHANGE
7707                            ;*      THEN THE READ DATA IS COMPARED
7708
7709                            ;;*****************************************************************
7710  030360  000004            TST41:  SCOPE
7711  030362  012706  001000            MOV     #STACK,SP       ;RESET STACK
7712  030366  012737  000041  004604    MOV     #41,@#TSTNM     ;SAVE TEST NUMBER
7713
7714  030374  004737  041524            JSR     PC,@#CLDISK     ;SET R1-RHCS1, R2-RHCS2
7715                                                            ;R3-RHDS1, R4-RHER1
7716                                                            ;GIVE RH-11 INITIALIZE
7717                                                            ;SETUP UNIT NUMBER
7718
7719                                    ;*FILL WRITE FROM BUFFER WITH HEADER
7720
7721  030400  004037  041350            JSR     R0,@#FLHEAD     ;SAVE HEADER DATA IN WRFROM
7722  030404  002470                    WRFROM                  ;LOCATION WHERE SAVED
7723  030406  000004                    4                       ;NUMBER OF WORDS SAVED
7724  030410  010000                    10000                   ;FIRST DATA WORD
7725  030412  000000                    0                       ;SECOND DATA WORD
7726  030414  000000                    0                       ;THIRD DATA WORD
7727  030416  000000                    0                       ;FOURTH DATA WORD
7728
7729                                    ;*FILL WRITE FROM BUFFER WITH DATA
7730  030420  004037  041374            JSR     R0,@#CLAREA     ;CLEAR 256. WORDS, FROM WRFROM+10
7731  030424  002500                    WRFROM+10                       ;STARTING FROM WRFROM+10
7732  030426  000400                    256.                    ;256. WORDS
7733  030430  000000                    0                       ;FILL WITH 0
7734
```

```
                                        ;*FILL WRITE FROM BUFFER WITH NEXT SECTOR HEADER
                                        
030432  004037  041350                  JSR     R0,@#FLHEAD     ;SAVE HEADER DATA IN WRFROM+<260.*2>
030436  003500                          WRFROM+<260.*2>                 ;LOCATION WHERE SAVED
030440  000004                          4                       ;NUMBER OF WORDS SAVED
030442  010000                          10000                   ;FIRST DATA WORD
030444  000001                          1                       ;SECOND DATA WORD
030446  000000                          0                       ;THIRD DATA WORD
030450  000000                          0                       ;FOURTH DATA WORD
                                        
                                        ;*FILL WRITE FROM BUFFER WITH NEXT SECTOR DATA
030452  004037  041374                  JSR     R0,@#CLAREA     ;CLEAR 2 WORDS, FROM WRFROM+<264.*2>
030456  003510                          WRFROM+<264.*2>                 ;STARTING FROM WRFROM+<264.*2>
030460  000002                          2                       ;2 WORDS
030462  000001                          1                       ;FILL WITH 1
                                        
                                        
                                        ;*NOW READ INTO BUFFER WILL BE FILLED WITH SAME DATA
                                        ;*AS WRITE FROM BUFFER SO THAT AFTER A WRITE COMPARISONS
                                        ;*CAN BE MADE TO MAKE SURE THAT WRITE DID NOT
                                        ;*CHANGE WRITE FROM BUFFER.
                                        
                                        
030464  004037  041350                  JSR     R0,@#FLHEAD     ;SAVE HEADER DATA IN REINTO
030470  003534                          REINTO                  ;LOCATION WHERE SAVED
030472  000004                          4                       ;NUMBER OF WORDS SAVED
030474  010000                          10000                   ;FIRST DATA WORD
030476  000000                          0                       ;SECOND DATA WORD
030500  000000                          0                       ;THIRD DATA WORD
030502  000000                          0                       ;FOURTH DATA WORD
030504  004037  041374                  JSR     R0,@#CLAREA     ;CLEAR 256. WORDS, FROM REINTO+10
030510  003544                          REINTO+10               ;STARTING FROM REINTO+10
030512  000400                          256.                    ;256. WORDS
030514  000000                          0                       ;FILL WITH 0
                                        
                                        
030516  004037  041350                  JSR     R0,@#FLHEAD     ;SAVE HEADER DATA IN REINTO+<260.*2>
030522  004544                          REINTO+<260.*2>                 ;LOCATION WHERE SAVED
030524  000004                          4                       ;NUMBER OF WORDS SAVED
030526  010000                          10000                   ;FIRST DATA WORD
030530  000001                          1                       ;SECOND DATA WORD
030532  000000                          0                       ;THIRD DATA WORD
030534  000000                          0                       ;FOURTH DATA WORD
030536  004037  041374                  JSR     R0,@#CLAREA     ;CLEAR 2 WORDS, FROM REINTO+<264.*2>
030542  004554                          REINTO+<264.*2>                 ;STARTING FROM REINTO+<264.*2>
030544  000002                          2                       ;2 WORDS
030546  000001                          1                       ;FILL WITH 1
                                        
                                        
                                        ;*NOW THE WRITE HEADER AND DATA COMMAND WILL BE FILLED
                                        
                                        
030550  004037  043476                  JSR     R0,@#RUN        ;SETUP TO RUN FOR DATA COMMAND
030554  000000                          0                       ;CYLINDER 0
030556     000                  .BYTE   0                       ;SECTOR 0
```

```
7791  030557  000                    .BYTE   0                       ;TRACK 0
7792  030560  177366                         -262.-4                 ;WORD COUNT (DATA) = 262. +
7793                                                                 ;4 HEADER WORDS
7794  030562  002470                         WRFROM                  ;BUS ADDRESS
7795                                                                 ;STARTING ADDRESS OF DATA
7796                                                                 ;BUFFER = WRFROM
7797  030564  000000                         0                       ;DO NOT INHIBIT BUS ADDRESS INCREMENT
7798  030566  010000                         FMT22                   ;16 BITS PER WORD FORMAT
7799                                                                 ;DO NOT INHIBIT ECC CORRECTION
7800                                                                 ;DO NOT INHIBIT HEADER COMPARE
7801  030570  002444                         WRIFOR                  ;GET READY TO DO A WRIFOR
7802                                                       ;WRITE HEADER AND DATA WITH 62 IN RHCS1
7803
7804
7805                                         ;*NOW SAVE REGISTERS FOR COMPARISON AFTER WRITE HEADER AND DAT1
7806  030572  004037  041672                 JSR     R0,@#SAVER      ;SAVE REGISTERS
7807  030576  002272                         RHWC                    ;RHWC IS THE FIRST REGISTER SAVED
7808  030600  004612                         SAVERE                  ;STARTING ADDRESS OF WHERE
7809                                                                 ;THE REGISTERS ARE SAVED
7810  030602  000022                         18.                     ;NUMBER OF REGISTERS
7811                                                                 ;SAVED = 18.
7812
7813  030604  004737  041604                 JSR     PC,@#CHECKT     ;CHECK DVA,RDY,MOL,DPR,DRY,VV = 1
7814                                                                 ;AND THAT NO STATUS BITS ARE STUCK = 1
7815  030610  104401  067033                 TYPE    ,CPHALT         ;CANNOT CONTINUE TESTING IF ANY OF
7816                                                                 ;THE FIRST SET OF BITS DON'T = 1
7817  030614  000000                         HALT                    ;STOP
7818
7819  030616  013777  004606  151442         MOV     @#RP4VEC,@RPVEC ;SET RP04 VECTOR ADDRESS
7820                                                                 ;TO 'TIME1' IF P-CLOCK IS PRESENT
7821                                                                 ;OR TO 'TIME2' IF P-CLOCK IS NOT PRESENT
7822                                                                 ;'TIME' WILL ONLY SAVE
7823                                                                 ;CURRENT CYLINDER ADDRESS
7824                                                                 ;AND LOOK AHEAD REGISTERS
7825
7826
7827  030624  013746  002444                 MOV     @#WRIFOR,-(SP)  ;GET READY TO MOVE COMMAND
7828  030630  052716  000101                 BIS     #GO!IE,(SP)     ;GET READY TO SET 'GO' AND
7829                                                                 ;ENABLE INTERRUPT
7830  030634  012677  151440                 MOV     (SP)+,@RHCS1    ;GO WITH
7831                                                                  ;62 IN RHCS1 FOR WRITE HEADER AND DATA
7832                                                                 ;WITH INTERRUPT ENABLED
7833  030640  011100                         MOV     @R1,R0          ;SAVE RHCS1 DURING ABOVE OPERATION
7834  030642  011305                         MOV     @R3,R5          ;SAVE RHDS1 DURING ABOVE OPERATION
7835
7836                                         ;*ONE REVOLUTION = 16670 MICRO SEC, ONE SECTOR = 760 MICRO SEC
7837
7838
7839  030644  104413                         WAT                     ;WAIT FOR RDY BIT TO SET
7840  030646  002300                         RHCS1                   ;WAIT FOR RHCS1 REGISTER
7841  030650  000200                         RDY                     ;WAIT FOR RDY BIT IN RHCS1 REGISTER
7842  030652  002001                         1025.                   ;ALLOW 10250 MICRO SECONDS
7843  030654  001553                         875.                    ;RDY MUST SET BETWEEN
7844                                                                 ;1500 AND 19000 MICRO SECONDS
7845
7846                                         ;*COMPARE CONTENTS OF RHCS1 AND RHDS1 ALREADY SAVED IN
```

```
7847                                        ;*R0 AND R5 IMMEDIATELY AFTER GO
7848
7849  030656  013746  002444                MOV     @#WRIFOR,-(SP)  ;SAVE COMMAND
7850  030662  052716  004101                BIS     #IE!GO!DVA,(SP) ;INCLUDE IE!GO!DVA
7851  030666  011637  001124                MOV     (SP),@#$GDDAT   ;SAVE FOR PRINTOUT
7852  030672  022600                        CMP     (SP)+,R0        ;DURING ABOVE OPERATION ONLY IE!GO!DVA
7853                                                                ;AND COMMAND SHOULD BE SET
7854  030674  001405                        BEQ     64$             ;BRANCH IF GOOD
7855  030676  010037  001126                MOV     R0,@#$BDDAT     ;BAD DATA
7856  030702  010137  004600                MOV     R1,@#REGADR     ;FAILING REGISTER RHCS1
7857  030706  104021                        ERROR   21              ;DURING ABOVE OPERATION ONLY
7858                                                                ;COMMAND AND IE!GO!DVA SHOULD BE SET
7859  030710  012746  010500        64$:    MOV     #MOL!DPR!VV,-(SP)       ;SAVE BITS SET DURING OPERATION IN RHDS1
7860  030714  011637  001124                MOV     (SP),@#$GDDAT   ;SAVE FOR PRINTOUT
7861  030720  022605                        CMP     (SP)+,R5        ;DURING ABOVE OPERATION ONLY MOL!DPR!VV
7862                                                                ;SHOULD BE SET
7863  030722  001405                        BEQ     66$             ;BRANCH IF GOOD
7864  030724  010537  001126                MOV     R5,@#$BDDAT     ;BAD DATA
7865  030730  010337  004600                MOV     R3,@#REGADR     ;FAILING REGISTER RHDS1
7866  030734  104063                        ERROR   63              ;DURING ABOVE OPERATION ONLY
7867                                                                ;MOL!DPR!VV SHOULD BE SET
7868  030736                        66$:
7869
7870                                        ;*NOW CHANGE SAVED REGISTERS TO EXPECTED VALUES
7871
7872  030736  004037  041426                JSR     R0,@#FILLRE     ;MOV 0 INTO SAVED RHWC
7873  030742  002272                        RHWC                    ;SAVED REGISTER TO CHANGE
7874  030744  000000                        0                       ;DATA
7875  030746  004037  041426                JSR     R0,@#FILLRE     ;MOV WRFROM+<266.*2> INTO SAVED RHBA
7876  030752  002274                        RHBA                    ;SAVED REGISTER TO CHANGE
7877  030754  003514                        WRFROM+<266.*2>                 ;DATA
7878  030756  004037  041426                JSR     R0,@#FILLRE     ;MOV 2 INTO SAVED RHDST
7879  030762  002304                        RHDST                   ;SAVED REGISTER TO CHANGE
7880  030764  000002                        2                       ;DATA
7881
7882                                        ;*NOW COMPARE REGISTERS BEFORE WRITE HEADER AND DATA
7883                                        ;*WITH REGISTERS AFTER COMMAND
7884
7885
7886  030766  004037  042512                JSR     R0,@#COMREG     ;COMPARE SAVED REGISTERS WITH
7887                                                                ;PRESENT VALUE
7888  030772  004612                        SAVERE                  ;GOOD DATA SAVED IN 'SAVERE'
7889  030774  002354                        WC                      ;TEST DATA STARTING FROM 'RHWC'
7890  030776  000022                        18.                     ;18. REGISTERS TO BE COMPARED
7891  031000  031004                        1$                      ;RETURN TO 1$ ON ERROR
7892  031002  031010                        2$                      ;RETURN TO 2$ ON NO ERROR
7893
7894  031004  104027                1$:     ERROR   27              ;WRITE HEADER AND DATA
7895  031006  000207                        RTS     PC              ;CAUSED IMPROPER REGISTER
7896                                                                ;CHANGE
7897                                                                ;GOOD DATA GIVES WHAT SHOULD
7898                                                                ;BE THERE
7899                                                                ;RECEIVED DATA GIVES WHAT
7900                                                                ;WAS THERE AFTER COMMANT
7901
7902                                        ;*NOW WRITE FROM BUFFER WILL BE CHECKED TO SEE THAT
```

```
                                        ;*NOTHING GOT CHANGED
                                
031010                          2$:
                                
031010  004037  043542                JSR     R0,@#COMPAR     ;COMPARE TWO BLOCKS OF MEMORY
031014  003534                        REINTO                  ;GOOD DATA STARTS FROM REINTO
031016  002470                        WRFROM                  ;TEST DATA STARTS FROM WRFROM
031020  000412                        266.                    ;266. WORDS TO BE COMPARED
031022  031026                        3$                      ;RETURN TO 3$ ON ERROR
031024  031032                        4$                      ;RETURN TO 4$ ON NO ERROR
                                
031026  104030                  3$:   ERROR   30              ;WRITE HEADER AND DATA
031030  000207                        RTS     PC              ;CHANGED WRITE FROM BUFFER
                                
                                        ;*NOW A READ HEADER AND DATA COMMAND WILL BE GIVEN
                                        ;*FOR SECTOR 1, 256 WORDS
                                        ;*READ INTO BUFFER IS FILLED WITH ONES
031032                          4$:
                                
031032  004737  041524                JSR     PC,@#CLDISK     ;SET R1-RHCS1, R2-RHCS2
                                                                ;R3-RHDS1, R4-RHER1
                                                                ;GIVE RH-11 INITIALIZE
                                                                ;SETUP UNIT NUMBER
031036  004037  041374                JSR     R0,@#CLAREA     ;CLEAR 260. WORDS, FROM REINTO
031042  003534                        REINTO                  ;STARTING FROM REINTO
031044  000404                        260.                    ;260. WORDS
031046  000377                        377                     ;FILL WITH 377
                                
                                        ;*WRITE FROM BUFFER IS FILLED WITH EXPECTED DATA
                                
031050  004037  041350                JSR     R0,@#FLHEAD     ;SAVE HEADER DATA IN WRFROM
031054  002470                        WRFROM                  ;LOCATION WHERE SAVED
031056  000006                        6                       ;NUMBER OF WORDS SAVED
031060  010000                        10000                   ;FIRST DATA WORD
031062  000001                        1                       ;SECOND DATA WORD
031064  000000                        0                       ;THIRD DATA WORD
031066  000000                        0                       ;FOURTH DATA WORD
031070  000001                        1                       ;FIFTH DATA WORD
031072  000001                        1                       ;SIXTH DATA WORD
031074  004037  041374                JSR     R0,@#CLAREA     ;CLEAR 254 WORDS, FROM WRFROM+<6.*2>
031100  002504                        WRFROM+<6.*2>                   ;STARTING FROM WRFROM+<6.*2>
031102  000254                        254                     ;254 WORDS
031104  000000                        0                       ;FILL WITH 0
                                
                                        ;*NOW FILL COMMAND
                                
031106  004037  043476                JSR     R0,@#RUN        ;SETUP TO RUN FOR DATA COMMAND
031112  000000                        0                       ;CYLINDER 0
031114     001                .BYTE   1                       ;SECTOR 1
031115     000                .BYTE   0                       ;TRACK 0
031116  177374                        -256.-4                 ;WORD COUNT (DATA) = 256. +
```

```
7959
7960  031120  003534                          REINTO                  ;4 HEADER WORDS
7961                                                                  ;BUS ADDRESS
7962                                                                  ;STARTING ADDRESS OF DATA
7963  031122  000000                          0                       ;BUFFER = REINTO
7964  031124  014000                          ECI!FMT22               ;DO NOT INHIBIT BUS ADDRESS INCREMENT
7965                                                                          ;16 BITS PER WORD FORMAT
7966                                                                  ;INHIBIT ECC CORRECTION
7967  031126  002450                          REFOR                   ;DO NOT INHIBIT HEADER COMPARE
7968                                                                  ;GET READY TO DO A REFOR
7969                                                                  ;READ HEADER AND DATA WITH 72 IN RHCS1
7970
7971                                          ;*NOW SAVE REGISTERS FOR COMPARISON AFTER READ HEADER AND DATA
7972
7973  031130  004037  041672                  JSR     R0,@#SAVER      ;SAVE REGISTERS
7974  031134  002272                          RHWC                    ;RHWC IS THE FIRST REGISTER SAVED
7975  031136  004612                          SAVERE                  ;STARTING ADDRESS OF WHERE
7976                                                                  ;THE REGISTERS ARE SAVED
7977  031140  000022                          18.                     ;NUMBER OF REGISTERS
7978                                                                  ;SAVED = 18.
7979
7980  031142  004737  041604                  JSR     PC,@#CHECKT     ;CHECK DVA,RDY,MOL,DPR,DRY,VV = 1
7981                                                                  ;AND THAT NO STATUS BITS ARE STUCK = 1
7982  031146  104401  067033                  TYPE    ,CPHALT         ;CANNOT CONTINUE TESTING IF ANY OF
7983                                                                  ;THE FIRST SET OF BITS DON'T = 1
7984  031152  000000                          HALT                    ;STOP
7985
7986  031154  013777  004606  151104          MOV     @#RP4VEC,@RPVEC ;SET RP04 VECTOR ADDRESS
7987                                                                  ;TO 'TIME1' IF P-CLOCK IS PRESENT
7988                                                                  ;OR TO 'TIME2' IF P-CLOCK IS NOT PRESENT
7989                                                                  ;'TIME' WILL ONLY SAVE
7990                                                                  ;CURRENT CYLINDER ADDRESS
7991                                                                  ;AND LOOK AHEAD REGISTERS
7992
7993
7994  031162  013746  002450                  MOV     @#REFOR,-(SP)   ;GET READY TO MOVE COMMAND
7995  031166  052716  000101                  BIS     #GO!IE,(SP)     ;GET READY TO SET 'GO' AND
7996                                                                  ;ENABLE INTERRUPT
7997  031172  012677  151102                  MOV     (SP)+,@RHCS1    ;GO WITH
7998                                                                    ;72 IN RHCS1 FOR READ DATA
7999                                                                  ;WITH INTERRUPT ENABLED
8000  031176  011100                          MOV     @R1,R0          ;SAVE RHCS1 DURING ABOVE OPERATION
8001  031200  011305                          MOV     @R3,R5          ;SAVE RHDS1 DURING ABOVE OPERATION
8002
8003                                          ;*ONE REVOLUTION = 16670 MICRO SEC, ONE SECTOR 760 MICRO SECONDS
8004
8005
8006  031202  104413                          WAT                     ;WAIT FOR RDY BIT TO SET
8007  031204  002300                          RHCS1                   ;WAIT FOR RHCS1 REGISTER
8008  031206  000200                          RDY                     ;WAIT FOR RDY BIT IN RHCS1 REGISTER
8009  031210  001614                          908.                    ;ALLOW 9080 MICRO SECONDS
8010  031212  001507                          839.                    ;RDY MUST SET BETWEEN
8011                                                                  ;690 AND 17470 MICRO SECONDS
8012
8013                                          ;*COMPARE CONTENTS OF RHCS1 AND RHDS1 ALREADY SAVED IN
8014                                          ;*R0 AND R5 IMMEDIATELY AFTER GO
```

```
8015
8016  031214  013746  002450                MOV     @#REFOR,-(SP)   ;SAVE COMMAND
8017  031220  052716  004101                BIS     #IE!GO!DVA,(SP) ;INCLUDE IE!GO!DVA
8018  031224  011637  001124                MOV     (SP),@#$GDDAT   ;SAVE FOR PRINTOUT
8019  031230  022600                       CMP     (SP)+,R0        ;DURING ABOVE OPERATION ONLY IE!GO!DVA
8020                                                                ;AND COMMAND SHOULD BE SET
8021  031232  001405                       BEQ     67$             ;BRANCH IF GOOD
8022  031234  010037  001126               MOV     R0,@#$BDDAT     ;BAD DATA
8023  031240  010137  004600               MOV     R1,@#REGADR     ;FAILING REGISTER RHCS1
8024  031244  104021                       ERROR   21              ;DURING ABOVE OPERATION ONLY
8025                                                                ;COMMAND AND IE!GO!DVA SHOULD BE SET
8026  031246  012746  010500        67$:   MOV     #MOL!DPR!VV,-(SP)       ;SAVE BITS SET DURING OPERATION IN RHDS1
8027  031252  011637  001124               MOV     (SP),@#$GDDAT   ;SAVE FOR PRINTOUT
8028  031256  022605                       CMP     (SP)+,R5        ;DURING ABOVE OPERATION ONLY MOL!DPR!VV
8029                                                                ;SHOULD BE SET
8030  031260  001405                       BEQ     69$             ;BRANCH IF GOOD
8031  031262  010537  001126               MOV     R5,@#$BDDAT     ;BAD DATA
8032  031266  010337  004600               MOV     R3,@#REGADR     ;FAILING REGISTER RHDS1
8033  031272  104063                       ERROR   63              ;DURING ABOVE OPERATION ONLY
8034                                                                ;MOL!DPR!VV SHOULD BE SET
8035  031274                       69$:
8036
8037                                       ;*CHANGE SAVED REGISTERS TO EXPECTED VALUES
8038
8039  031274  004037  041426               JSR     R0,@#FILLRE     ;MOV 0 INTO SAVED RHWC
8040  031300  002272                       RHWC                    ;SAVED REGISTER TO CHANGE
8041  031302  000000                       0                       ;DATA
8042  031304  004037  041426               JSR     R0,@#FILLRE     ;MOV REINTO+<260.*2> INTO SAVED RHBA
8043  031310  002274                       RHBA                    ;SAVED REGISTER TO CHANGE
8044  031312  004544                       REINTO+<260.*2>                 ;DATA
8045  031314  004037  041426               JSR     R0,@#FILLRE     ;MOV 2 INTO SAVED RHDST
8046  031320  002304                       RHDST                   ;SAVED REGISTER TO CHANGE
8047  031322  000002                       2                       ;DATA
8048
8049                                       ;*COMPARE REGISTERS BEFORE READ HEADER AND DATA
8050                                       ;*WITH REGISTERS AFTER COMMAND
8051
8052
8053  031324  004037  042512               JSR     R0,@#COMREG     ;COMPARE SAVED REGISTERS WITH
8054                                                                ;PRESENT VALUE
8055  031330  004612                       SAVERE                  ;GOOD DATA SAVED IN 'SAVERE'
8056  031332  002354                       WC                      ;TEST DATA STARTING FROM 'RHWC'
8057  031334  000022                       18.                     ;18. REGISTERS TO BE COMPARED
8058  031336  031342                       5$                      ;RETURN TO 5$ ON ERROR
8059  031340  031346                       6$                      ;RETURN TO 6$ ON NO ERROR
8060
8061
8062  031342  104031               5$:     ERROR   31              ;READ HEADER AND DATA CAUSED
8063  031344  000207                       RTS     PC              ;IMPROPER REGISTER CHANGE
8064                                                                ;GOOD DATA GIVES WHAT SHOULD
8065                                                                ;BE THERE
8066                                                                ;RECEIVED DATA GIVES WHAT WAS
8067                                                                ;THERE AFTER COMMAND
8068
8069                                       ;*NOW READ INTO BUFFER WILL BE CHECKED TO SEE
8070                                       ;*THAT READ WAS GOOD
```

```
 8071
 8072  031346                          6$:
 8073
 8074  031346  004037  043542                JSR     R0,@#COMPAR     ;COMPARE TWO BLOCKS OF MEMORY
 8075  031352  002470                        WRFROM                  ;GOOD DATA STARTS FROM WRFROM
 8076  031354  003534                        REINTO                  ;TEST DATA STARTS FROM REINTO
 8077  031356  000404                        260.                    ;260. WORDS TO BE COMPARED
 8078  031360  031364                        7$                      ;RETURN TO 7$ ON ERROR
 8079  031362  031370                        10$                     ;RETURN TO 10$ ON NO ERROR
 8080
 8081
 8082
 8083  031364  104032                  7$:   ERROR   32              ;WRITE HEADER AND DATA
 8084  031366  000207                        RTS     PC              ;FOLLOWED BY A READ HEADER
 8085                                                                ;AND DATA GAVE A READ ERROR
 8086                                                                ;ERROR MAY BE IN READ OR WRITE
 8087  031370                          10$:
```

```
8088
8089                            ;;*****************************************************************
8090                            ;*TEST 42       IMPLIED SEEK TO CYL 001
8091
8092                                    .SBTTL  SEEK TESTS
8093
8094                            ;*      ONLY NEW ADDITION IN THIS TEST IS AN IMPLIED SEEK FROM CYLINDER 0
8095                            ;*      TO CYLINDER 1, A WRITE HEADER AND DATA IS GIVEN FOR MORE THAN
8096                            ;*      ONE SECTOR OF WORDS
8097                            ;*
8098                            ;*
8099                            ;*      WRITE HEADER AND DATA CYLINDER 0, FORMAT 16 BITS PER WORD
8100                            ;*      TRACK 18, SECTOR 21, KEYS=0, NUMBER OF WORDS 266 WORDS
8101                            ;*      (4 HEADER, 256 DATA=1125, 4 HEADER 2 DATA = 2000
8102                            ;*      THEN READ HEADER AND DATA FOR ABOVE, TRACK 0, SECTOR 0, CYL=1
8103                            ;*      WRITE FROM BUFFER AND READ INTO BUFFER ARE FILLED WITH
8104                            ;*      10000,0,0,0 AND 256 OF 1125, 10001,0,0,0, AND 2 OF 2000
8105                            ;*      THE WRITE COMMAND IS THEN LOADED INTO THE REGISTERS EXCEPT
8106                            ;*      THE GO BIT, AND ALL THE REGISTERS ARE SAVED
8107                            ;*      THEN GO IS GIVEN FOR WRITE HEADER AND DATA
8108                            ;*
8109                            ;*      THEN ALL REGISTERS ARE COMPARED TO CHECK FOR IMPROPER CHANGE
8110                            ;*      THEN WRITE FROM BUFFER IS CHECKED TO SEE THAT NOTHING CHANGED
8111                            ;*
8112                            ;*
8113                            ;*      NOW FOR READ COMMAND READ INTO BUFFER IS FILLED
8114                            ;*      WITH ALL ONES, WRITE FROM BUFFER IS LOADED WITH
8115                            ;*      10001,0,0,0 AND 2 OF 2000,254 OF ZEROS COMMAND IS LOADED
8116                            ;*      INTO REGISTERS EXCEPT GO AND IE ALL REGISTERS ARE SAVED
8117                            ;*      GO IS GIVEN FOR THE READ COMMAND
8118                            ;*
8119                            ;*      ALL REGISTERS ARE CHECKED FOR IMPROPER CHANGE
8120                            ;*      THEN THE READ DATA IS COMPARED.
8121
8122                            ;;*****************************************************************
8123  031370  000004            TST42:  SCOPE
8124  031372  012706  001000            MOV     #STACK,SP       ;RESET STACK
8125  031376  012737  000042  004604    MOV     #42,@#TSTNM     ;SAVE TEST NUMBER
8126
8127  031404  004737  041524            JSR     PC,@#CLDISK     ;SET R1-RHCS1, R2-RHCS2
8128                                                            ;R3-RHDS1, R4-RHER1
8129                                                            ;GIVE RH-11 INITIALIZE
8130                                                            ;SETUP UNIT NUMBER
8131
8132                                    ;*FILL WRITE FROM BUFFER WITH HEADER
8133
8134  031410  004037  041350            JSR     R0,@#FLHEAD     ;SAVE HEADER DATA IN WRFROM
8135  031414  002470                    WRFROM                  ;LOCATION WHERE SAVED
8136  031416  000004                    4                       ;NUMBER OF WORDS SAVED
8137  031420  010000                    10000                   ;FIRST DATA WORD
8138  031422  011025                    <18.*400>!<21.>                 ;SECOND DATA WORD
8139  031424  000000                    0                       ;THIRD DATA WORD
8140  031426  000000                    0                       ;FOURTH DATA WORD
8141
8142                                    ;*FILL WRITE FROM BUFFER WITH DATA
8143  031430  004037  041374            JSR     R0,@#CLAREA     ;CLEAR 256. WORDS, FROM WRFROM+10
```

```
                                        WRFROM+10                       ;STARTING FROM WRFROM+10
                                        256.                    ;256. WORDS
                                        <18.*40>!21.                    ;FILL WITH <18.*40>!21.


                                        ;*FILL WRITE FROM BUFFER WITH NEXT TRACK HEADER

031442  004037  041350                  JSR     R0,@#FLHEAD     ;SAVE HEADER DATA IN WRFROM+<260.*2>
031446  003500                          WRFROM+<260.*2>                 ;LOCATION WHERE SAVED
031450  000004                          4                       ;NUMBER OF WORDS SAVED
031452  010001                          10001                   ;FIRST DATA WORD
031454  000000                          0                       ;SECOND DATA WORD
031456  000000                          0                       ;THIRD DATA WORD
031460  000000                          0                       ;FOURTH DATA WORD

                                        ;*FILL WRITE FROM BUFFER WITH NEXT TRACK DATA
031462  004037  041374                  JSR     R0,@#CLAREA     ;CLEAR 2 WORDS, FROM WRFROM+<264.*2>
031466  003510                          WRFROM+<264.*2>                 ;STARTING FROM WRFROM+<264.*2>
031470  000002                          2                       ;2 WORDS
031472  002000                          2000                    ;FILL WITH 2000


                                        ;*NOW READ INTO BUFFER WILL BE FILLED WITH SAME DATA
                                        ;*AS WRITE FROM BUFFER SO THAT AFTER A WRITE COMPARISONS
                                        ;*CAN BE MADE TO MAKE SURE THAT WRITE DID NOT
                                        ;*CHANGE WRITE FROM BUFFER.


031474  004037  041350                  JSR     R0,@#FLHEAD     ;SAVE HEADER DATA IN REINTO
031500  003534                          REINTO                  ;LOCATION WHERE SAVED
031502  000004                          4                       ;NUMBER OF WORDS SAVED
031504  010000                          10000                   ;FIRST DATA WORD
031506  011025                          <18.*400>!<21.>                 ;SECOND DATA WORD
031510  000000                          0                       ;THIRD DATA WORD
031512  000000                          0                       ;FOURTH DATA WORD
031514  004037  041374                  JSR     R0,@#CLAREA     ;CLEAR 256. WORDS, FROM REINTO+10
031520  003544                          REINTO+10                       ;STARTING FROM REINTO+10
031522  000400                          256.                    ;256. WORDS
031524  001125                          <18.*40>!21.                    ;FILL WITH <18.*40>!21.


031526  004037  041350                  JSR     R0,@#FLHEAD     ;SAVE HEADER DATA IN REINTO+<260.*2>
031532  004544                          REINTO+<260.*2>                 ;LOCATION WHERE SAVED
031534  000004                          4                       ;NUMBER OF WORDS SAVED
031536  010001                          10001                   ;FIRST DATA WORD
031540  000000                          0                       ;SECOND DATA WORD
031542  000000                          0                       ;THIRD DATA WORD
031544  000000                          0                       ;FOURTH DATA WORD
031546  004037  041374                  JSR     R0,@#CLAREA     ;CLEAR 2 WORDS, FROM REINTO+<264.*2>
031552  004554                          REINTO+<264.*2>                 ;STARTING FROM REINTO+<264.*2>
031554  000002                          2                       ;2 WORDS
031556  002000                          2000                    ;FILL WITH 2000


                                        ;*NOW THE WRITE HEADER AND DATA COMMAND WILL BE FILLED
```

```
8200
8201  031560  004037  043476              JSR     R0,@#RUN        ;SETUP TO RUN FOR DATA COMMAND
8202  031564  000000                      0                       ;CYLINDER 0
8203  031566     025                .BYTE  21.                     ;SECTOR 21.
8204  031567     022                .BYTE  18.                     ;TRACK 18.
8205  031570  177366                      -262.-4                 ;WORD COUNT (DATA) = 262. +
8206                                                              ;4 HEADER WORDS
8207  031572  002470                      WRFROM                  ;BUS ADDRESS
8208                                                              ;STARTING ADDRESS OF DATA
8209                                                              ;BUFFER = WRFROM
8210  031574  000000                      0                       ;DO NOT INHIBIT BUS ADDRESS INCREMENT
8211  031576  010000                      FMT22                   ;16 BITS PER WORD FORMAT
8212                                                              ;DO NOT INHIBIT ECC CORRECTION
8213                                                              ;DO NOT INHIBIT HEADER COMPARE
8214  031600  002444                      WRIFOR                  ;GET READY TO DO A WRIFOR
8215                                                    ;WRITE HEADER AND DATA WITH 62 IN RHCS1
8216
8217
8218                                      ;*NOW SAVE REGISTERS FOR COMPARISON AFTER WRITE HEADER AND DATA1
8219
8220  031602  004037  041672              JSR     R0,@#SAVER      ;SAVE REGISTERS
8221  031606  002272                      RHWC                    ;RHWC IS THE FIRST REGISTER SAVED
8222  031610  004612                      SAVERE                  ;STARTING ADDRESS OF WHERE
8223                                                              ;THE REGISTERS ARE SAVED
8224  031612  000022                      18.                     ;NUMBER OF REGISTERS
8225                                                              ;SAVED = 18.
8226
8227  031614  004737  041604              JSR     PC,@#CHECKT     ;CHECK DVA,RDY,MOL,DPR,DRY,VV = 1
8228                                                              ;AND THAT NO STATUS BITS ARE STUCK = 1
8229  031620  104401  067033              TYPE    ,CPHALT         ;CANNOT CONTINUE TESTING IF ANY OF
8230                                                              ;THE FIRST SET OF BITS DON'T = 1
8231  031624  000000                      HALT                    ;STOP
8232
8233  031626  013777  004606  150432      MOV     @#RP4VEC,@RPVEC ;SET RP04 VECTOR ADDRESS
8234                                                              ;TO 'TIME1' IF P-CLOCK IS PRESENT
8235                                                              ;OR TO 'TIME2' IF P-CLOCK IS NOT PRESENT
8236                                                              ;'TIME' WILL ONLY SAVE
8237                                                              ;CURRENT CYLINDER ADDRESS
8238                                                              ;AND LOOK AHEAD REGISTERS
8239
8240
8241  031634  013746  002444              MOV     @#WRIFOR,-(SP)  ;GET READY TO MOVE COMMAND
8242  031640  052716  000101              BIS     #GO!IE,(SP)     ;GET READY TO SET 'GO' AND
8243                                                              ;ENABLE INTERRUPT
8244  031644  012677  150430              MOV     (SP)+,@RHCS1    ;GO WITH
8245                                                               ;62 IN RHCS1 FOR WRITE HEADER AND DATA
8246                                                              ;WITH INTERRUPT ENABLED
8247  031650  011100                      MOV     @R1,R0          ;SAVE RHCS1 DURING ABOVE OPERATION
8248  031652  011305                      MOV     @R3,R5          ;SAVE RHDS1 DURING ABOVE OPERATION
8249
8250                                      ;*ONE REVOLUTION = 16670 MICRO1 SEC, ONE SECTOR = 760 MICRO1 SEC
8251
8252
8253  031654  104413                      WAT                     ;WAIT FOR RDY BIT TO SET
8254  031656  002300                      RHCS1                   ;WAIT FOR RHCS1 REGISTER
8255  031660  000200                      RDY                     ;WAIT FOR RDY BIT IN RHCS1 REGISTER
```

```
8256  031662  003237                        1695.                   ;ALLOW 16950 MICRO SECONDS
8257  031664  001515                        845.                    ;RDY MUST SET BETWEEN
8258                                                                  ;8500 AND 25400 MICRO SECONDS
8259
8260                                        ;*COMPARE CONTENTS OF RHCS1 AND RHDS1 ALREADY SAVED IN
8261                                        ;*R0 AND R5 IMMEDIATELY AFTER GO
8262
8263  031666  013746  002444                MOV     @#WRIFOR,-(SP)  ;SAVE COMMAND
8264  031672  052716  004101                BIS     #IE!GO!DVA,(SP) ;INCLUDE IE!GO!DVA
8265  031676  011637  001124                MOV     (SP),@#$GDDAT   ;SAVE FOR PRINTOUT
8266  031702  022600                        CMP     (SP)+,R0        ;DURING ABOVE OPERATION ONLY IE!GO!DVA
8267                                                                ;AND COMMAND SHOULD BE SET
8268  031704  001405                        BEQ     64$             ;BRANCH IF GOOD
8269  031706  010037  001126                MOV     R0,@#$BDDAT     ;BAD DATA
8270  031712  010137  004600                MOV     R1,@#REGADR     ;FAILING REGISTER RHCS1
8271  031716  104021                        ERROR   21              ;DURING ABOVE OPERATION ONLY
8272                                                                ;COMMAND AND IE!GO!DVA SHOULD BE SET
8273  031720  012746  010500        64$:    MOV     #MOL!DPR!VV,-(SP)       ;SAVE BITS SET DURING OPERATION IN RHDS1
8274  031724  011637  001124                MOV     (SP),@#$GDDAT   ;SAVE FOR PRINTOUT
8275  031730  022605                        CMP     (SP)+,R5        ;DURING ABOVE OPERATION ONLY MOL!DPR!VV
8276                                                                ;SHOULD BE SET
8277  031732  001405                        BEQ     66$             ;BRANCH IF GOOD
8278  031734  010537  001126                MOV     R5,@#$BDDAT     ;BAD DATA
8279  031740  010337  004600                MOV     R3,@#REGADR     ;FAILING REGISTER RHDS1
8280  031744  104063                        ERROR   63              ;DURING ABOVE OPERATION ONLY
8281                                                                ;MOL!DPR!VV SHOULD BE SET
8282  031746                        66$:
8283
8284                                        ;*NOW CHANGE SAVED REGISTERS TO EXPECTED VALUES
8285
8286  031746  004037  041426                JSR     R0,@#FILLRE     ;MOV 0 INTO SAVED RHWC
8287  031752  002272                        RHWC                    ;SAVED REGISTER TO CHANGE
8288  031754  000000                        0                       ;DATA
8289  031756  004037  041426                JSR     R0,@#FILLRE     ;MOV WRFROM+<266.*2> INTO SAVED RHBA
8290  031762  002274                        RHBA                    ;SAVED REGISTER TO CHANGE
8291  031764  003514                        WRFROM+<266.*2>                 ;DATA
8292  031766  004037  041426                JSR     R0,@#FILLRE     ;MOV 1 INTO SAVED RHCA
8293  031772  002312                        RHCA                    ;SAVED REGISTER TO CHANGE
8294  031774  000001                        1                       ;DATA
8295  031776  004037  041426                JSR     R0,@#FILLRE     ;MOV 1 INTO SAVED RHCC
8296  032002  002334                        RHCC                    ;SAVED REGISTER TO CHANGE
8297  032004  000001                        1                       ;DATA
8298  032006  004037  041426                JSR     R0,@#FILLRE     ;MOV 1 INTO SAVED RHDST
8299  032012  002304                        RHDST                   ;SAVED REGISTER TO CHANGE
8300  032014  000001                        1                       ;DATA
8301
8302                                        ;*NOW COMARE REGISTERS BEFORE WRITE HEADER AND DATA
8303                                        ;*WITH REGISTERS AFTER COMMAND
8304
8305
8306  032016  004037  042512                JSR     R0,@#COMREG     ;COMPARE SAVED REGISTERS WITH
8307                                                                ;PRESENT VALUE
8308  032022  004612                        SAVERE                  ;GOOD DATA SAVED IN 'SAVERE'
8309  032024  002354                        WC                      ;TEST DATA STARTING FROM 'RHWC'
8310  032026  000022                        18.                     ;18. REGISTERS TO BE COMPARED
8311  032030  032034                        1$                      ;RETURN TO 1$ ON ERROR
```

```
8312  032032  032040                  2$                      ;RETURN TO 2$ ON NO ERROR
8313
8314  032034  104027          1$:     ERROR   27              ;WRITE HEADER AND DATA
8315  032036  000207                  RTS     PC              ;CAUSED IMPROPER REGISTER
8316                                                          ;CHANGE
8317                                                          ;GOOD DATA GIVES WHAT SHOULD
8318                                                          ;BE THERE
8319                                                          ;RECEIVED DATA GIVES WHAT
8320                                                          ;WAS THERE AFTER COMMANT
8321
8322                                  ;*NOW WRITE FROM BUFFER WILL BE CHECKED TO SEE THAT
8323                                  ;*NOTHING GOT CHANGED
8324
8325  032040                  2$:
8326
8327  032040  004037  043542          JSR     R0,@#COMPAR     ;COMPARE TWO BLOCKS OF MEMORY
8328  032044  003534                  REINTO                  ;GOOD DATA STARTS FROM REINTO
8329  032046  002470                  WRFROM                  ;TEST DATA STARTS FROM WRFROM
8330  032050  000412                  266.                    ;266. WORDS TO BE COMPARED
8331  032052  032056                  3$                      ;RETURN TO 3$ ON ERROR
8332  032054  032062                  4$                      ;RETURN TO 4$ ON NO ERROR
8333
8334
8335  032056  104030          3$:     ERROR   30              ;WRITE HEADER AND DATA
8336  032060  000207                  RTS     PC              ;CHANGED WRITE FROM BUFFER
8337
8338                                  ;*NOW A READ HEADER AND DATA COMMAND WILL BE GIVEN
8339                                  ;*READ INTO BUFFER IS FILLED WITH ONES
8340
8341  032062                  4$:
8342
8343  032062  004737  041524          JSR     PC,@#CLDISK     ;SET R1-RHCS1, R2-RHCS2
8344                                                          ;R3-RHDS1, R4-RHER1
8345                                                          ;GIVE RH-11 INITIALIZE
8346                                                          ;SETUP UNIT NUMBER
8347  032066  004037  041374          JSR     R0,@#CLAREA     ;CLEAR 260. WORDS, FROM REINTO
8348  032072  003534                  REINTO                  ;STARTING FROM REINTO
8349  032074  000404                  260.                    ;260. WORDS
8350  032076  177777                  -1                      ;FILL WITH -1
8351
8352
8353                                  ;*WRITE FROM BUFFER IS FILLED WITH 10001,0,0,0,2000,2000, AND 254 OF 0
8354
8355  032100  004037  041350          JSR     R0,@#FLHEAD     ;SAVE HEADER DATA IN WRFROM
8356  032104  002470                  WRFROM                  ;LOCATION WHERE SAVED
8357  032106  000006                  6                       ;NUMBER OF WORDS SAVED
8358  032110  010001                  10001                   ;FIRST DATA WORD
8359  032112  000000                  0                       ;SECOND DATA WORD
8360  032114  000000                  0                       ;THIRD DATA WORD
8361  032116  000000                  0                       ;FOURTH DATA WORD
8362  032120  002000                  2000                    ;FIFTH DATA WORD
8363  032122  002000                  2000                    ;SIXTH DATA WORD
8364  032124  004037  041374          JSR     R0,@#CLAREA     ;CLEAR 254. WORDS, FROM WRFROM+<6*2>
8365  032130  002504                  WRFROM+<6*2>                    ;STARTING FROM WRFROM+<6*2>
8366  032132  000376                  254.                    ;254. WORDS
8367  032134  000000                  0                       ;FILL WITH 0
```

```
                                        ;*NOW FILL COMMAND
                                        ;
032136  004037  043476                  JSR     R0,@#RUN        ;SETUP TO RUN FOR DATA COMMAND
032142  000001                          1                       ;CYLINDER 1
032144     000                .BYTE     0                       ;SECTOR 0
032145     000                .BYTE     0                       ;TRACK 0
032146  177374                          -256.-4                 ;WORD COUNT (DATA) = 256. +
                                                                ;4 HEADER WORDS
032150  003534                          REINTO                  ;BUS ADDRESS
                                                                ;STARTING ADDRESS OF DATA
                                                                ;BUFFER = REINTO
032152  000000                          0                       ;DO NOT INHIBIT BUS ADDRESS INCREMENT
032154  014000                          ECI!FMT22                       ;16 BITS PER WORD FORMAT
                                                                ;INHIBIT ECC CORRECTION
                                                                ;DO NOT INHIBIT HEADER COMPARE
032156  002450                          REFOR                   ;GET READY TO DO A REFOR
                                                                ;READ HEADER AND DATA WITH 72 IN RHCS1
                                        ;
                                        ;*NOW SAVE REGISTERS FOR COMPARISON AFTER READ HEADER AND DATA
                                        ;
032160  004037  041672                  JSR     R0,@#SAVER      ;SAVE REGISTERS
032164  002272                          RHWC                    ;RHWC IS THE FIRST REGISTER SAVED
032166  004612                          SAVERE                  ;STARTING ADDRESS OF WHERE
                                                                ;THE REGISTERS ARE SAVED
032170  000022                          18.                     ;NUMBER OF REGISTERS
                                                                ;SAVED = 18.
                                        ;
032172  004737  041604          JSR     PC,@#CHECKT             ;CHECK DVA,RDY,MOL,DPR,DRY,VV = 1
                                                                ;AND THAT NO STATUS BITS ARE STUCK = 1
032176  104401  067033          TYPE    ,CPHALT                 ;CANNOT CONTINUE TESTING IF ANY OF
                                                                ;THE FIRST SET OF BITS DON'T = 1
032202  000000                  HALT                            ;STOP
                                        ;
032204  013777  004606  150054  MOV     @#RP4VEC,@RPVEC ;SET RP04 VECTOR ADDRESS
                                                                ;TO 'TIME1' IF P-CLOCK IS PRESENT
                                                                ;OR TO 'TIME2' IF P-CLOCK IS NOT PRESENT
                                                                ;'TIME' WILL ONLY SAVE
                                                                ;CURRENT CYLINDER ADDRESS
                                                                ;AND LOOK AHEAD REGISTERS
                                        ;
                                        ;
032212  013746  002450          MOV     @#REFOR,-(SP)   ;GET READY TO MOVE COMMAND
032216  052716  000101          BIS     #GO!IE,(SP)     ;GET READY TO SET 'GO' AND
                                                        ;ENABLE INTERRUPT
032222  012677  150052          MOV     (SP)+,@RHCS1    ;GO WITH
                                                          ;72 IN RHCS1 FOR READ DATA
                                                        ;WITH INTERRUPT ENABLED
032226  011100                  MOV     @R1,R0          ;SAVE RHCS1 DURING ABOVE OPERATION
032230  011305                  MOV     @R3,R5          ;SAVE RHDS1 DURING ABOVE OPERATION
                                        ;
032232  104413                  WAT                     ;WAIT FOR RDY BIT TO SET
032234  002300                  RHCS1                   ;WAIT FOR RHCS1 REGISTER
```

```
8424  032236  000200                          RDY                     ;WAIT FOR RDY BIT IN RHCS1 REGISTER
8425  032240  001614                          908.                    ;ALLOW 9080 MICRO SECONDS
8426  032242  001507                          839.                    ;RDY MUST SET BETWEEN
8427                                                                  ;690 AND 17470 MICRO SECONDS
8428
8429                                          ;*COMPARE CONTENTS OF RHCS1 AND RHDS1 ALREADY SAVED IN
8430                                          ;*R0 AND R5 IMMEDIATELY AFTER GO
8431
8432  032244  013746  002450                  MOV     @#REFOR,-(SP)   ;SAVE COMMAND
8433  032250  052716  004101                  BIS     #IE!GO!DVA,(SP) ;INCLUDE IE!GO!DVA
8434  032254  011637  001124                  MOV     (SP),@#$GDDAT   ;SAVE FOR PRINTOUT
8435  032260  022600                          CMP     (SP)+,R0        ;DURING ABOVE OPERATION ONLY IE!GO!DVA
8436                                                                  ;AND COMMAND SHOULD BE SET
8437  032262  001405                          BEQ     67$             ;BRANCH IF GOOD
8438  032264  010037  001126                  MOV     R0,@#$BDDAT     ;BAD DATA
8439  032270  010137  004600                  MOV     R1,@#REGADR     ;FAILING REGISTER RHCS1
8440  032274  104021                          ERROR   21              ;DURING ABOVE OPERATION ONLY
8441                                                                  ;COMMAND AND IE!GO!DVA SHOULD BE SET
8442  032276  012746  010500          67$:    MOV     #MOL!DPR!VV,-(SP)       ;SAVE BITS SET DURING OPERATION IN RHDS1
8443  032302  011637  001124                  MOV     (SP),@#$GDDAT   ;SAVE FOR PRINTOUT
8444  032306  022605                          CMP     (SP)+,R5        ;DURING ABOVE OPERATION ONLY MOL!DPR!VV
8445                                                                  ;SHOULD BE SET
8446  032310  001405                          BEQ     69$             ;BRANCH IF GOOD
8447  032312  010537  001126                  MOV     R5,@#$BDDAT     ;BAD DATA
8448  032316  010337  004600                  MOV     R3,@#REGADR     ;FAILING REGISTER RHDS1
8449  032322  104063                          ERROR   63              ;DURING ABOVE OPERATION ONLY
8450                                                                  ;MOL!DPR!VV SHOULD BE SET
8451  032324                          69$:
8452
8453                                          ;*CHANGE SAVED REGISTERS TO EXPECTED VALUES
8454  032324  004037  041426                  JSR     R0,@#FILLRE     ;MOV 0 INTO SAVED RHWC
8455  032330  002272                          RHWC                    ;SAVED REGISTER TO CHANGE
8456  032332  000000                          0                       ;DATA
8457  032334  004037  041426                  JSR     R0,@#FILLRE     ;MOV REINTO+<260.*2> INTO SAVED RHBA
8458  032340  002274                          RHBA                    ;SAVED REGISTER TO CHANGE
8459  032342  004544                          REINTO+<260.*2>                 ;DATA
8460  032344  004037  041426                  JSR     R0,@#FILLRE     ;MOV 1 INTO SAVED RHDST
8461  032350  002304                          RHDST                   ;SAVED REGISTER TO CHANGE
8462  032352  000001                          1                       ;DATA
8463
8464                                          ;*COMPARE REGISTERS BEFORE READ HEADER AND DATA
8465                                          ;*WITH REGISTERS AFTER COMMAND
8466
8467  032354  004037  042512                  JSR     R0,@#COMREG     ;COMPARE SAVED REGISTERS WITH
8468                                                                  ;PRESENT VALUE
8469  032360  004612                          SAVERE                  ;GOOD DATA SAVED IN 'SAVERE'
8470  032362  002354                          WC                      ;TEST DATA STARTING FROM 'RHWC'
8471  032364  000022                          18.                     ;18. REGISTERS TO BE COMPARED
8472  032366  032372                          5$                      ;RETURN TO 5$ ON ERROR
8473  032370  032376                          6$                      ;RETURN TO 6$ ON NO ERROR
8474
8475
8476  032372  104031                  5$:     ERROR   31              ;READ HEADER AND DATA CAUSED
8477  032374  000207                          RTS     PC              ;IMPROPER REGISTER CHANGE
8478                                                                  ;GOOD DATA GIVES WHAT SHOULD
8479                                                                  ;BE THERE
```

```
                                                        ;RECEIVED DATA GIVES WHAT WAS
                                                        ;THERE AFTER COMMAND

                                    ;*NOW READ INTO BUFFER WILL BE CHECKED TO SEE
                                    ;*THAT READ WAS GOOD

032376                          6$:

032376  004037  043542                JSR     R0,@#COMPAR     ;COMPARE TWO BLOCKS OF MEMORY
032402  002470                    WRFROM                      ;GOOD DATA STARTS FROM WRFROM
032404  003534                    REINTO                      ;TEST DATA STARTS FROM REINTO
032406  000404                    260.                        ;260. WORDS TO BE COMPARED
032410  032414                    7$                          ;RETURN TO 7$ ON ERROR
032412  032420                    10$                         ;RETURN TO 10$ ON NO ERROR


032414  104032                7$:     ERROR   32              ;WRITE HEADER AND DATA
032416  000207                        RTS     PC              ;FOLLOWED BY A READ HEADER
                                                        ;AND DATA GAVE A READ ERROR
                                                        ;ERROR MAY BE IN READ OR WRITE
032420                          10$:
```

```
                        ;;*****************************************************************
                        ;*TEST 43       SEEK & WRT TEST (CYL = 0-10)

                        ;*      THIS TEST GETS THE HEADS OUT TO CYLINDER 10 NOT BY ONE
                        ;*      SEEK BUT BY TEN IMPLIED SEEKS ONE CYLINDER AT A TIME
                        ;*
                        ;*      THIS TEST STARTS WITH A (ALREADY TESTED) RECALIBRATE
                        ;*      THAT IS CYLINDER ZERO.  THEN ON CYLINDER 0 SECTOR
                        ;*      #21 TRACK #18 IT WRITES 266 WORDS THERE BY GETTING
                        ;*      THE HEAD TO CYLINDER 1 THEN IT WRITES 266 WORDS ON
                        ;*      CYLINDER 1 SECTOR #21 TRACK #18 THERE BY GETTING
                        ;*      THE HEADS TO CYLINDER 2
                        ;*      THIS IS REPEATED 10 TIMES GETTING THE
                        ;*      HEADS TO CYLINDER 10
                        ;*      THEN A SEEK COMMAND IS GIVEN TO CYLINDER 0
                        ;*      AND DATA ALREADY WRITTEN IS CHECKED

                        ;;*****************************************************************
032420  000004          TST43:  SCOPE
032422  012706  001000          MOV     #STACK,SP       ;RESET STACK
032426  012737  000043  004604  MOV     #43,@#TSTNM     ;SAVE TEST NUMBER

032434  004737  041524          JSR     PC,@#CLDISK     ;SET R1-RHCS1, R2-RHCS2
                                                        ;R3-RHDS1, R4-RHER1
                                                        ;GIVE RH-11 INITIALIZE
                                                        ;SETUP UNIT NUMBER

                        ;*THE FOLLOWING MOVES ARE TO INITIALIZE TEST FROM
                        ;*CYLINDER 0
                        ;*THESE LOCATIONS ARE CHANGED DURING TEST TO ENABLE
                        ;*GOING TO NEXT CYLINDER

032440  012737  010000  032616  MOV     #10000,@#ST1+10
032446  012737  001125  032636  MOV     #<<18.*40>!21.>,@#ST2+10
032454  012737  010001  032650  MOV     #10001,@#ST3+10
032462  012737  002000  032670  MOV     #2000,@#ST4+10
032470  012737  000000  032676  MOV     #0,@#ST5+4
032476  012737  000001  033106  MOV     #1,@#ST6+6
032504  012737  000001  033116  MOV     #1,@#ST6+16
032512  012737  010001  033200  MOV     #10001,@#ST9+10
032520  012737  002000  033220  MOV     #2000,@#ST10+10
032526  012737  000001  033240  MOV     #1,@#ST11+4

                        ;*THIS IS TO GET THE HEADS TO CYLINDER 0

032534  013777  004606  147524  MOV     @#RP4VEC,@RPVEC ;SET RP04 VECTOR ADDRESS
                                                        ;TO 'TIME1' IF P-CLOCK IS PRESENT
                                                        ;OR TO 'TIME2' IF P-CLOCK IS NOT PRESENT
                                                        ;'TIME' WILL ONLY SAVE
                                                        ;CURRENT CYLINDER ADDRESS
                                                        ;AND LOOK AHEAD REGISTERS

032542  013746  002426          MOV     @#RECALI,-(SP)  ;GET READY TO MOVE COMMAND
032546  052716  000101          BIS     #GO!IE,(SP)     ;GET READY TO SET 'GO' AND
                                                        ;ENABLE INTERRUPT
```

```
8559  032552  012677  147522              MOV     (SP)+,@RHCS1    ;GO WITH
8560                                                              ;6 IN RHCS1 FOR RECALIBRATE
8561                                                              ;WITH INTERRUPT ENABLED
8562  032556  011100                      MOV     @R1,R0          ;SAVE RHCS1 DURING ABOVE OPERATION
8563  032560  011305                      MOV     @R3,R5          ;SAVE RHDS1 DURING ABOVE OPERATION
8564
8565  032562  104413                      WAT                     ;WAIT FOR DRY BIT TO SET
8566  032564  002322                      RHDS1                   ;WAIT FOR RHDS1 REGISTER
8567  032566  000200                      DRY                     ;WAIT FOR DRY BIT IN RHDS1 REGISTER
8568  032570  076377                      31999.                  ;ALLOW 319990 MICRO SECONDS
8569  032572  056701                      24001.                  ;DRY MUST SET BETWEEN
8570                                                              ;79980 AND 560000 MICRO SECONDS
8571  032574  012737  000012  001200      MOV     #10.,@#$TMP1    ;TEN COUNT TO GET TO CYLINDER 10
8572
8573  032602  004737  041524              JSR     PC,@#CLDISK     ;SET R1-RHCS1, R2-RHCS2
8574                                                              ;R3-RHDS1, R4-RHER1
8575                                                              ;GIVE RH-11 INITIALIZE
8576                                                              ;SETUP UNIT NUMBER
8577
8578                                      ;*FILL WRITE FROM BUFFER WITH HEADER
8579  032606                      ST1:
8580
8581  032606  004037  041350              JSR     R0,@#FLHEAD     ;SAVE HEADER DATA IN WRFROM
8582  032612  002470                      WRFROM                  ;LOCATION WHERE SAVED
8583  032614  000004                      4                       ;NUMBER OF WORDS SAVED
8584  032616  010000                      10000                   ;FIRST DATA WORD
8585  032620  011025                      <18.*400>!21.                   ;SECOND DATA WORD
8586  032622  000000                      0                       ;THIRD DATA WORD
8587  032624  000000                      0                       ;FOURTH DATA WORD
8588
8589                                      ;*FILL WRITE FROM BUFFER WITH DATA
8590  032626                      ST2:
8591  032626  004037  041374              JSR     R0,@#CLAREA     ;CLEAR 256. WORDS, FROM WRFROM+10
8592  032632  002500                      WRFROM+10               ;STARTING FROM WRFROM+10
8593  032634  000400                      256.                    ;256. WORDS
8594  032636  001125                      <0.*2000>!<18.*40>!21.                  ;FILL WITH <0.*2000>!<18.*40>!21.
8595
8596
8597                                      ;*FILL WRITE FROM BUFFER WITH NEXT TRACK HEADER
8598  032640                      ST3:
8599
8600  032640  004037  041350              JSR     R0,@#FLHEAD     ;SAVE HEADER DATA IN WRFROM+<260.*2>
8601  032644  003500                      WRFROM+<260.*2>                 ;LOCATION WHERE SAVED
8602  032646  000004                      4                       ;NUMBER OF WORDS SAVED
8603  032650  010001                      10001                   ;FIRST DATA WORD
8604  032652  000000                      0                       ;SECOND DATA WORD
8605  032654  000000                      0                       ;THIRD DATA WORD
8606  032656  000000                      0                       ;FOURTH DATA WORD
8607
8608                                      ;*FILL WRITE FROM BUFFER WITH NEXT TRACK DATA
8609  032660                      ST4:
8610  032660  004037  041374              JSR     R0,@#CLAREA     ;CLEAR 2 WORDS, FROM WRFROM+<264.*2>
8611  032664  003510                      WRFROM+<264.*2>                 ;STARTING FROM WRFROM+<264.*2>
8612  032666  000002                      2                       ;2 WORDS
8613  032670  002000                      1.*2000                 ;FILL WITH 1.*2000
8614
```

```
                                                ;*THE WRITE HEADER AND DATA COMMAND WILL BE FILLED
032672                                  ST5:
                                                JSR     R0,@#RUN        ;SETUP TO RUN FOR DATA COMMAND
032672  004037  043476                          0                       ;CYLINDER 0
032676  000000                          .BYTE   21.                     ;SECTOR 21.
032700     025                          .BYTE   18.                     ;TRACK 18.
032701     022                                  -262.-4                 ;WORD COUNT (DATA) = 262. +
032702  177366                                                          ;4 HEADER WORDS
                                                WRFROM                  ;BUS ADDRESS
032704  002470                                                          ;STARTING ADDRESS OF DATA
                                                                        ;BUFFER = WRFROM
                                                0                       ;DO NOT INHIBIT BUS ADDRESS INCREMENT
032706  000000                                  FMT22                   ;16 BITS PER WORD FORMAT
032710  010000                                                          ;DO NOT INHIBIT ECC CORRECTION
                                                                        ;DO NOT INHIBIT HEADER COMPARE
                                                WRIFOR                  ;GET READY TO DO A WRIFOR
032712  002444                                                  ;WRITE HEADER AND DATA WITH 62 IN RHCS1


                                                ;*SAVE REGISTERS FOR COMPARISON AFTER WRITE HEADER AND DATA

                                                JSR     R0,@#SAVER      ;SAVE REGISTERS
032714  004037  041672                          RHWC                    ;RHWC IS THE FIRST REGISTER SAVED
032720  002272                                  SAVERE                  ;STARTING ADDRESS OF WHERE
032722  004612                                                          ;THE REGISTERS ARE SAVED
                                                18.                     ;NUMBER OF REGISTERS
032724  000022                                                          ;SAVED = 18.


                                                JSR     PC,@#CHECKT     ;CHECK DVA,RDY,MOL,DPR,DRY,VV = 1
032726  004737  041604                                                  ;AND THAT NO STATUS BITS ARE STUCK = 1
                                                TYPE    ,CPHALT         ;CANNOT CONTINUE TESTING IF ANY OF
032732  104401  067033                                                  ;THE FIRST SET OF BITS DON'T = 1
                                                HALT                    ;STOP
032736  000000


                                                MOV     @#RP4VEC,@RPVEC ;SET RP04 VECTOR ADDRESS
032740  013777  004606  147320                                          ;TO 'TIME1' IF P-CLOCK IS PRESENT
                                                                        ;OR TO 'TIME2' IF P-CLOCK IS NOT PRESENT
                                                                        ;'TIME' WILL ONLY SAVE
                                                                        ;CURRENT CYLINDER ADDRESS
                                                                        ;AND LOOK AHEAD REGISTERS


                                                MOV     @#WRIFOR,-(SP)  ;GET READY TO MOVE COMMAND
032746  013746  002444                          BIS     #GO!IE,(SP)     ;GET READY TO SET 'GO' AND
032752  052716  000101                                                  ;ENABLE INTERRUPT
                                                MOV     (SP)+,@RHCS1    ;GO WITH
032756  012677  147316                                                   ;62 IN RHCS1 FOR WRITE HEADER AND DATA
                                                                        ;WITH INTERRUPT ENABLED
                                                MOV     @R1,R0          ;SAVE RHCS1 DURING ABOVE OPERATION
032762  011100                                  MOV     @R3,R5          ;SAVE RHDS1 DURING ABOVE OPERATION
032764  011305

                                                ;*ONE REVOLUTION = 16670 MICRO SECONDS, ONE SECTOR = 760 MICRO SEC.
                                                ;*MAX TIME ALLOWED = ONE REVOLUTION + SEEK + 2 SECTORS
                                                ;*MIN TIME ALLOWED = 2 SECTORS + SEEK
```

```
8671
8672
8673  032766  104413                          WAT                     ;WAIT FOR RDY BIT TO SET
8674  032770  002300                          RHCS1                   ;WAIT FOR RHCS1 REGISTER
8675  032772  000200                          RDY                     ;WAIT FOR RDY BIT IN RHCS1 REGISTER
8676  032774  003237                          1695.                   ;ALLOW 16950 MICRO SECONDS
8677  032776  001515                          845.                    ;RDY MUST SET BETWEEN
8678                                                                  ;8500 AND 25400 MICRO SECONDS
8679
8680                                          ;*COMPARE CONTENTS OF RHCS1 AND RHDS1 ALREADY SAVED IN
8681                                          ;*R0 AND R5 IMMEDIATELY AFTER GO
8682
8683  033000  013746  002444                  MOV     @#WRIFOR,-(SP)  ;SAVE COMMAND
8684  033004  052716  004101                  BIS     #IE!GO!DVA,(SP) ;INCLUDE IE!GO!DVA
8685  033010  011637  001124                  MOV     (SP),@#$GDDAT   ;SAVE FOR PRINTOUT
8686  033014  022600                          CMP     (SP)+,R0        ;DURING ABOVE OPERATION ONLY IE!GO!DVA
8687                                                                  ;AND COMMAND SHOULD BE SET
8688  033016  001405                          BEQ     64$             ;BRANCH IF GOOD
8689  033020  010037  001126                  MOV     R0,@#$BDDAT     ;BAD DATA
8690  033024  010137  004600                  MOV     R1,@#REGADR     ;FAILING REGISTER RHCS1
8691  033030  104021                          ERROR   21              ;DURING ABOVE OPERATION ONLY
8692                                                                  ;COMMAND AND IE!GO!DVA SHOULD BE SET
8693  033032  012746  010500          64$:    MOV     #MOL!DPR!VV,-(SP)       ;SAVE BITS SET DURING OPERATION IN RHDS1
8694  033036  011637  001124                  MOV     (SP),@#$GDDAT   ;SAVE FOR PRINTOUT
8695  033042  022605                          CMP     (SP)+,R5        ;DURING ABOVE OPERATION ONLY MOL!DPR!VV
8696                                                                  ;SHOULD BE SET
8697  033044  001405                          BEQ     66$             ;BRANCH IF GOOD
8698  033046  010537  001126                  MOV     R5,@#$BDDAT     ;BAD DATA
8699  033052  010337  004600                  MOV     R3,@#REGADR     ;FAILING REGISTER RHDS1
8700  033056  104063                          ERROR   63              ;DURING ABOVE OPERATION ONLY
8701                                                                  ;MOL!DPR!VV SHOULD BE SET
8702  033060                          66$:
8703
8704                                          ;*NOW CHANGES SAVED REGISTERS TO EXPECTED VALUES
8705
8706  033060  004037  041426                  JSR     R0,@#FILLRE     ;MOV 0 INTO SAVED RHWC
8707  033064  002272                          RHWC                    ;SAVED REGISTER TO CHANGE
8708  033066  000000                          0                       ;DATA
8709  033070  004037  041426                  JSR     R0,@#FILLRE     ;MOV WRFROM+<266.*2> INTO SAVED RHBA
8710  033074  002274                          RHBA                    ;SAVED REGISTER TO CHANGE
8711  033076  003514                          WRFROM+<266.*2>                 ;DATA
8712  033100                          ST6:
8713  033100  004037  041426                  JSR     R0,@#FILLRE     ;MOV 1 INTO SAVED RHCC
8714  033104  002334                          RHCC                    ;SAVED REGISTER TO CHANGE
8715  033106  000001                          1                       ;DATA
8716  033110  004037  041426                  JSR     R0,@#FILLRE     ;MOV 1 INTO SAVED RHCA
8717  033114  002312                          RHCA                    ;SAVED REGISTER TO CHANGE
8718  033116  000001                          1                       ;DATA
8719  033120  004037  041426                  JSR     R0,@#FILLRE     ;MOV 1 INTO SAVED RHDST
8720  033124  002304                          RHDST                   ;SAVED REGISTER TO CHANGE
8721  033126  000001                          1                       ;DATA
8722
8723                                          ;*COMPARE REGISTERS BEFORE WRITE HEADER AND DATA
8724                                          ;*WITH REGISTERS AFTER COMMAND
8725
8726
```

```
8727  033130  004037  042512                JSR     R0,@#COMREG     ;COMPARE SAVED REGISTERS WITH
8728                                                                  ;PRESENT VALUE
8729  033134  004612                        SAVERE                  ;GOOD DATA SAVED IN 'SAVERE'
8730  033136  002354                        WC                      ;TEST DATA STARTING FROM 'RHWC'
8731  033140  000022                        18.                     ;18. REGISTERS TO BE COMPARED
8732  033142  033146                        ST7                     ;RETURN TO ST7 ON ERROR
8733  033144  033152                        ST8                     ;RETURN TO ST8 ON NO ERROR
8734  033146  104027                ST7:    ERROR   27              ;WRITE HEADER AND DATA CAUSED
8735  033150  000207                        RTS     PC              ;IMPROPER REGISTER CHANGE
8736                                                                  ;GOOD DATA GIVES WHAT SHOULD BE
8737                                                                  ;THERE
8738                                                                  ;RECEIVED DATA GIVES WHAT WAS BE
8739                                                                  ;THERE AFTER COMMAND
8740
8741                                        ;*SETUP TO READ HEADER AND DATA FOR NEXT TRACK
8742                                        ;*FILL READ INTO BUFFER WITH ALL ONES
8743
8744  033152                        ST8:
8745
8746  033152  004737  041524                JSR     PC,@#CLDISK     ;SET R1-RHCS1, R2-RHCS2
8747                                                                  ;R3-RHDS1, R4-RHER1
8748                                                                  ;GIVE RH-11 INITIALIZE
8749                                                                  ;SETUP UNIT NUMBER
8750  033156  004037  041374                JSR     R0,@#CLAREA     ;CLEAR 260. WORDS, FROM REINTO
8751  033162  003534                        REINTO                  ;STARTING FROM REINTO
8752  033164  000404                        260.                    ;260. WORDS
8753  033166  177777                        -1                      ;FILL WITH -1
8754
8755
8756                                        ;*FILL WRITE FROM BUFFER WITH EXPECTED DATA
8757
8758  033170                        ST9:
8759
8760  033170  004037  041350                JSR     R0,@#FLHEAD     ;SAVE HEADER DATA IN WRFROM
8761  033174  002470                        WRFROM                  ;LOCATION WHERE SAVED
8762  033176  000004                        4                       ;NUMBER OF WORDS SAVED
8763  033200  010001                        10001                   ;FIRST DATA WORD
8764  033202  000000                        0                       ;SECOND DATA WORD
8765  033204  000000                        0                       ;THIRD DATA WORD
8766  033206  000000                        0                       ;FOURTH DATA WORD
8767  033210                        ST10:
8768  033210  004037  041374                JSR     R0,@#CLAREA     ;CLEAR 2 WORDS, FROM WRFROM+<4*2>
8769  033214  002500                        WRFROM+<4*2>                    ;STARTING FROM WRFROM+<4*2>
8770  033216  000002                        2                       ;2 WORDS
8771  033220  002000                        1*2000                  ;FILL WITH 1*2000
8772
8773  033222  004037  041374                JSR     R0,@#CLAREA     ;CLEAR 254. WORDS, FROM WRFROM+<6*2>
8774  033226  002504                        WRFROM+<6*2>                    ;STARTING FROM WRFROM+<6*2>
8775  033230  000376                        254.                    ;254. WORDS
8776  033232  000000                        0                       ;FILL WITH 0
8777
8778
8779                                        ;*FILL COMMAND INTO REGISTERS
8780  033234                        ST11:
8781
8782  033234  004037  043476                JSR     R0,@#RUN        ;SETUP TO RUN FOR DATA COMMAND
```

```
 8783  033240  000001                      1                       ;CYLINDER 1
 8784  033242     000              .BYTE   0                       ;SECTOR 0
 8785  033243     000              .BYTE   0                       ;TRACK 0
 8786  033244  177374                      -256.-4                 ;WORD COUNT (DATA) = 256. +
 8787                                                              ;4 HEADER WORDS
 8788  033246  003534                      REINTO                  ;BUS ADDRESS
 8789                                                              ;STARTING ADDRESS OF DATA
 8790                                                              ;BUFFER = REINTO
 8791  033250  000000                      0                       ;DO NOT INHIBIT BUS ADDRESS INCREMENT
 8792  033252  014000                      ECI!FMT22                       ;16 BITS PER WORD FORMAT
 8793                                                              ;INHIBIT ECC CORRECTION
 8794                                                              ;DO NOT INHIBIT HEADER COMPARE
 8795  033254  002450                      REFOR                   ;GET READY TO DO A REFOR
 8796                                                              ;READ HEADER AND DATA WITH 72 IN RHCS1
 8797
 8798
 8799                                      ;*SAVE REGISTERS FOR COMPARISON AFTER READ HEADER
 8800                                      ;*AND DATA
 8801
 8802  033256  004037  041672              JSR     R0,@#SAVER      ;SAVE REGISTERS
 8803  033262  002272                      RHWC                    ;RHWC IS THE FIRST REGISTER SAVED
 8804  033264  004612                      SAVERE                  ;STARTING ADDRESS OF WHERE
 8805                                                              ;THE REGISTERS ARE SAVED
 8806  033266  000020                      16.                     ;NUMBER OF REGISTERS
 8807                                                              ;SAVED = 16.
 8808
 8809  033270  004737  041604              JSR     PC,@#CHECKT     ;CHECK DVA,RDY,MOL,DPR,DRY,VV = 1
 8810                                                              ;AND THAT NO STATUS BITS ARE STUCK = 1
 8811  033274  104401  067033              TYPE    ,CPHALT         ;CANNOT CONTINUE TESTING IF ANY OF
 8812                                                              ;THE FIRST SET OF BITS DON'T = 1
 8813  033300  000000                      HALT                    ;STOP
 8814
 8815  033302  013777  004606  146756      MOV     @#RP4VEC,@RPVEC ;SET RP04 VECTOR ADDRESS
 8816                                                              ;TO 'TIME1' IF P-CLOCK IS PRESENT
 8817                                                              ;OR TO 'TIME2' IF P-CLOCK IS NOT PRESENT
 8818                                                              ;'TIME' WILL ONLY SAVE
 8819                                                              ;CURRENT CYLINDER ADDRESS
 8820                                                              ;AND LOOK AHEAD REGISTERS
 8821
 8822
 8823  033310  013746  002450              MOV     @#REFOR,-(SP)   ;GET READY TO MOVE COMMAND
 8824  033314  052716  000101              BIS     #GO!IE,(SP)     ;GET READY TO SET 'GO' AND
 8825                                                              ;ENABLE INTERRUPT
 8826  033320  012677  146754              MOV     (SP)+,@RHCS1    ;GO WITH
 8827                                                                ;72 IN RHCS1 FOR READ DATA
 8828                                                              ;WITH INTERRUPT ENABLED
 8829  033324  011100                      MOV     @R1,R0          ;SAVE RHCS1 DURING ABOVE OPERATION
 8830  033326  011305                      MOV     @R3,R5          ;SAVE RHDS1 DURING ABOVE OPERATION
 8831
 8832
 8833  033330  104413                      WAT                     ;WAIT FOR RDY BIT TO SET
 8834  033332  002300                      RHCS1                   ;WAIT FOR RHCS1 REGISTER
 8835  033334  000200                      RDY                     ;WAIT FOR RDY BIT IN RHCS1 REGISTER
 8836  033336  001614                      908.                    ;ALLOW 9080 MICRO SECONDS
 8837  033340  001507                      839.                    ;RDY MUST SET BETWEEN
 8838                                                              ;690 AND 17470 MICRO SECONDS
```

```
                                        ;*COMPARE CONTENTS OF RHCS1 AND RHDS1 ALREADY SAVED IN
                                        ;*R0 AND R5 IMMEDIATELY AFTER GO

033342  013746  002450                  MOV     @#REFOR,-(SP)   ;SAVE COMMAND
033346  052716  004101                  BIS     #IE!GO!DVA,(SP) ;INCLUDE IE!GO!DVA
033352  011637  001124                  MOV     (SP),@#$GDDAT   ;SAVE FOR PRINTOUT
033356  022600                          CMP     (SP)+,R0        ;DURING ABOVE OPERATION ONLY IE!GO!DVA
                                                                ;AND COMMAND SHOULD BE SET
033360  001405                          BEQ     64$             ;BRANCH IF GOOD
033362  010037  001126                  MOV     R0,@#$BDDAT     ;BAD DATA
033366  010137  004600                  MOV     R1,@#REGADR     ;FAILING REGISTER RHCS1
033372  104021                          ERROR   21              ;DURING ABOVE OPERATION ONLY
                                                                ;COMMAND AND IE!GO!DVA SHOULD BE SET
033374  012746  010500          64$:    MOV     #MOL!DPR!VV,-(SP)       ;SAVE BITS SET DURING OPERATION IN RHDS1
033400  011637  001124                  MOV     (SP),@#$GDDAT   ;SAVE FOR PRINTOUT
033404  022605                          CMP     (SP)+,R5        ;DURING ABOVE OPERATION ONLY MOL!DPR!VV
                                                                ;SHOULD BE SET
033406  001405                          BEQ     66$             ;BRANCH IF GOOD
033410  010537  001126                  MOV     R5,@#$BDDAT     ;BAD DATA
033414  010337  004600                  MOV     R3,@#REGADR     ;FAILING REGISTER RHDS1
033420  104063                          ERROR   63              ;DURING ABOVE OPERATION ONLY
                                                                ;MOL!DPR!VV SHOULD BE SET
033422                          66$:

                                        ;*CHANGE SAVED REGISTERS TO EXPECTED VALUES

033422  004037  041426                  JSR     R0,@#FILLRE     ;MOV 0 INTO SAVED RHWC
033426  002272                          RHWC                    ;SAVED REGISTER TO CHANGE
033430  000000                          0                       ;DATA
033432  004037  041426                  JSR     R0,@#FILLRE     ;MOV REINTO+<260.*2> INTO SAVED RHBA
033436  002274                          RHBA                    ;SAVED REGISTER TO CHANGE
033440  004544                          REINTO+<260.*2>                 ;DATA
033442  004037  041426                  JSR     R0,@#FILLRE     ;MOV 1 INTO SAVED RHDST
033446  002304                          RHDST                   ;SAVED REGISTER TO CHANGE
033450  000001                          1                       ;DATA

                                        ;*COMPARE REGISTERS BEFORE READ HEADER AND DATA WITH
                                        ;*REGISTERS AFTER COMMAND


033452  004037  042512                  JSR     R0,@#COMREG     ;COMPARE SAVED REGISTERS WITH
                                                                ;PRESENT VALUE
033456  004612                          SAVERE                  ;GOOD DATA SAVED IN 'SAVERE'
033460  002354                          WC                      ;TEST DATA STARTING FROM 'RHWC'
033462  000022                          18.                     ;18. REGISTERS TO BE COMPARED
033464  033470                          ST12                    ;RETURN TO ST12 ON ERROR
033466  033474                          ST13                    ;RETURN TO ST13 ON NO ERROR

033470  104031          ST12:           ERROR   31              ;READ HEADER AND DATA CAUSED
033472  000207                          RTS     PC              ;IMPROPER REGISTER CHANGE
                                                                ;GOOD DATA GIVES WHAT SHOULD
                                                                ;BE THERE
                                                                ;RECEIVED DATA GIVES WHAT
                                                                ;WAS THERE AFTER COMMAND
```

```
                                        ;*READ INTO BUFFER IS CHECKED FOR PROPER READ
033474                          ST13:
033474  004037  043542                  JSR     R0,@#COMPAR     ;COMPARE TWO BLOCKS OF MEMORY
033500  002470                          WRFROM                  ;GOOD DATA STARTS FROM WRFROM
033502  003534                          REINTO                  ;TEST DATA STARTS FROM REINTO
033504  000404                          260.                    ;260. WORDS TO BE COMPARED
033506  033512                          ST14                    ;RETURN TO ST14 ON ERROR
033510  033516                          ST15                    ;RETURN TO ST15 ON NO ERROR

033512  104032                  ST14:   ERROR   32              ;WRITE HEADER AND DATA
033514  000207                          RTS     PC              ;WITH AN IMPLIED SEEK
                                                                ;FOLLOWED BY A READ
                                                                ;HEADER AND DATA ON THE
                                                                ;NEXT TRACK GAVE A
                                                                ;READ ERROR
                                                                ;ERROR MAY BE READ OR WRITE

                                        ;*THE HEADS HAVE ADVANCED ONE CYLINDER BY AN IMPLIED
                                        ;*SEEK
                                        ;*CHANGES WILL BE MADE TO ENABLE GOING TO THE NEXT
                                        ;*CYLINDER AND THEN THE ABOVE WILL BE REPEATED
                                        ;*TILL CYLINDER 10 IS REACHED

033516  005237  032616          ST15:   INC     @#ST1+10
033522  062737  002000  032636          ADD     #<1.*2000>,@#ST2+10
033530  005237  032650                  INC     @#ST3+10
033534  062737  002000  032670          ADD     #<1.*2000>,@#ST4+10
033542  005237  032676                  INC     @#ST5+4
033546  005237  033106                  INC     @#ST6+6
033552  005237  033116                  INC     @#ST6+16
033556  005237  033200                  INC     @#ST9+10
033562  062737  002000  033220          ADD     #<1.*2000>,@#ST10+10
033570  005237  033240                  INC     @#ST11+4
033574  005337  001200                  DEC     @#$TMP1         ;COUNT FOR TEN TIMES
033600  001001                          BNE     ST16            ;BRANCH IF 10 NOT DONE
033602  000402                          BR      ST17            ;10 COMPLETED SO CONTINUE
033604  000137  032606          ST16:   JMP     ST1             ;JUMP AS 10 NOT DONE

                                        ;*THE HEADS ARE NOW AT CYLINDER 10
                                        ;*ALL REGISTERS WILL BE SAVED AND A SEEK WILL BE GIVEN
                                        ;*TO CYLINDER 0
                                        ;*FILL REGISTERS FOR A SEEK COMMAND

033610                          ST17:

033610  004737  041524                  JSR     PC,@#CLDISK     ;SET R1-RHCS1, R2-RHCS2
                                                                ;R3-RHDS1, R4-RHER1
                                                                ;GIVE RH-11 INITIALIZE
                                                                ;SETUP UNIT NUMBER

033614  004037  041474                  JSR     R0,@#SEEKCY     ;SEEK FOR
033620  000000                          0                       ;CYLINDER 0

                                        ;*SAVE REGISTERS FOR COMPARISON AFTER SEEK COMMAND
```

```
 8951  033622  004037  041672                  JSR     R0,@#SAVER      ;SAVE REGISTERS
 8952  033626  002272                          RHWC                    ;RHWC IS THE FIRST REGISTER SAVED
 8953  033630  004612                          SAVERE                  ;STARTING ADDRESS OF WHERE
 8954                                                                  ;THE REGISTERS ARE SAVED
 8955  033632  000022                          18.                     ;NUMBER OF REGISTERS
 8956                                                                  ;SAVED = 18.
 8957
 8958  033634  013777  004606  146424          MOV     @#RP4VEC,@RPVEC ;SET RP04 VECTOR ADDRESS
 8959                                                                  ;TO 'TIME1' IF P-CLOCK IS PRESENT
 8960                                                                  ;OR TO 'TIME2' IF P-CLOCK IS NOT PRESENT
 8961                                                                  ;'TIME' WILL ONLY SAVE
 8962                                                                  ;CURRENT CYLINDER ADDRESS
 8963                                                                  ;AND LOOK AHEAD REGISTERS
 8964
 8965
 8966  033642  013746  002452                  MOV     @#SEECOM,-(SP)  ;GET READY TO MOVE COMMAND
 8967  033646  052716  000101                  BIS     #GO!IE,(SP)     ;GET READY TO SET 'GO' AND
 8968                                                                  ;ENABLE INTERRUPT
 8969  033652  012677  146422                  MOV     (SP)+,@RHCS1    ;GO WITH
 8970                                                                   ;4 IN RHCS1 FOR SEEK
 8971                                                                  ;WITH INTERRUPT ENABLED
 8972  033656  011100                          MOV     @R1,R0          ;SAVE RHCS1 DURING ABOVE OPERATION
 8973  033660  011305                          MOV     @R3,R5          ;SAVE RHDS1 DURING ABOVE OPERATION
 8974
 8975                                          ;*SEEK FOR ONE CYLINDER=7MILI SEC., FOR TEN=70 MILI SEC
 8976
 8977  033662  104413                          WAT                     ;WAIT FOR DRY BIT TO SET
 8978  033664  002322                          RHDS1                   ;WAIT FOR RHDS1 REGISTER
 8979  033666  000200                          DRY                     ;WAIT FOR DRY BIT IN RHDS1 REGISTER
 8980  033670  015530                          7000.                   ;ALLOW 70000 MICRO SECONDS
 8981  033672  000043                          35.                     ;DRY MUST SET BETWEEN
 8982                                                                  ;69650 AND 70350 MICRO SECONDS
 8983
 8984                                          ;*COMPARE CONTENTS OF RHCS1 AND RHDS1 ALREADY SAVED IN
 8985                                          ;*R0 AND R5 IMMEDIATELY AFTER GO
 8986
 8987  033674  013746  002452                  MOV     @#SEECOM,-(SP)  ;SAVE COMMAND
 8988  033700  052716  004301                  BIS     #DVA!GO!IE!RDY,(SP)     ;INCLUDE DVA!GO!IE!RDY
 8989  033704  011637  001124                  MOV     (SP),@#$GDDAT   ;SAVE FOR PRINTOUT
 8990  033710  022600                          CMP     (SP)+,R0        ;DURING ABOVE OPERATION ONLY DVA!GO!IE!RDY
 8991                                                                  ;AND COMMAND SHOULD BE SET
 8992  033712  001405                          BEQ     64$             ;BRANCH IF GOOD
 8993  033714  010037  001126                  MOV     R0,@#$BDDAT     ;BAD DATA
 8994  033720  010137  004600                  MOV     R1,@#REGADR     ;FAILING REGISTER RHCS1
 8995  033724  104021                          ERROR   21              ;DURING ABOVE OPERATION ONLY
 8996                                                                  ;COMMAND AND DVA!GO!IE!RDY SHOULD BE SET
 8997  033726  012746  030500          64$:    MOV     #PIP!MOL!DPR!VV,-(SP)   ;SAVE BITS SET DURING OPERATION IN RHDS1
 8998  033732  011637  001124                  MOV     (SP),@#$GDDAT   ;SAVE FOR PRINTOUT
 8999  033736  022605                          CMP     (SP)+,R5        ;DURING ABOVE OPERATION ONLY PIP!MOL!DPR!VV
 9000                                                                  ;SHOULD BE SET
 9001  033740  001405                          BEQ     66$             ;BRANCH IF GOOD
 9002  033742  010537  001126                  MOV     R5,@#$BDDAT     ;BAD DATA
 9003  033746  010337  004600                  MOV     R3,@#REGADR     ;FAILING REGISTER RHDS1
 9004  033752  104063                          ERROR   63              ;DURING ABOVE OPERATION ONLY
 9005                                                                  ;PIP!MOL!DPR!VV SHOULD BE SET
 9006  033754                          66$:
```

```
                                        ;*NOW CHANGE SAVED REGISTERS TO EXPECTED VALUE

033754  004037  041426               JSR     R0,@#FILLRE     ;MOV 0 INTO SAVED RHCC
033760  002334                       RHCC                    ;SAVED REGISTER TO CHANGE
033762  000000                       0                       ;DATA

033764  053737  004740  004636       BIS     @#ATTENT,@#SAVERE+24    ;SET APPROPRIATE 'ATA' BITS
                                                             ;FOR WORKING DRIVE IN
                                                             ;SAVED RHAS LOACTION

033772  004037  042404               JSR     R0,@#CHREG      ;CHANGE BITS IN SAVED REGISTER
033776  002322                       RHDS1                   ;CHANGE RHDS1 REGISTER

034000  000001                       1                       ;1 BIT/BITS TO BE CHANGED
034002  000001                       1                       ;NEW VALUE OF ATA IS 1
034004  100000                       ATA                     ;CHANGE ATA BIT

034006  004037  042404               JSR     R0,@#CHREG      ;CHANGE BITS IN SAVED REGISTER
034012  002300                       RHCS1                   ;CHANGE RHCS1 REGISTER

034014  000001                       1                       ;1 BIT/BITS TO BE CHANGED
034016  000001                       1                       ;NEW VALUE OF SC IS 1
034020  100000                       SC                      ;CHANGE SC BIT

                                        ;*COMPARE REGISTERS AFTER A SEEK COMMAND

034022  004037  042512               JSR     R0,@#COMREG     ;COMPARE SAVED REGISTERS WITH
                                                             ;PRESENT VALUE
034026  004612                       SAVERE                  ;GOOD DATA SAVED IN 'SAVERE'
034030  002354                       WC                      ;TEST DATA STARTING FROM 'RHWC'
034032  000022                       18.                     ;18. REGISTERS TO BE COMPARED
034034  034040                       ST18                    ;RETURN TO ST18 ON ERROR
034036  034044                       ST19                    ;RETURN TO ST19 ON NO ERROR

034040  104037          ST18:        ERROR   37              ;SEEK COMMAND CAUSED AN
034042  000207                       RTS     PC              ;ERROR
                                                             ;GOOD DATA GIVES WHAT SHOULD
                                                             ;BE THERE
                                                             ;RECEIVED DATA GIVES WHAT WAS
                                                             ;THERE AFTER A SEEK COMMAND

                                        ;*AT THIS POINT THE CURRENT CYLINDER IS GOOD AND THERE ARE
                                        ;*NO ERROR BITS
                                        ;*A READ HEADER AND DATA WILL BE DONE ON CYLINDER 0
                                        ;*SECTOR 21 TRACK 18, EXPECTED DATA IS 1125
                                        ;*FOR 10 WORDS
                                        ;*CLEAR READ INTO BUFFER WITH ALL ONES

034044          ST19:
034044  004037  041374               JSR     R0,@#CLAREA     ;CLEAR 260. WORDS, FROM REINTO
034050  003534                       REINTO                  ;STARTING FROM REINTO
034052  000404                       260.                    ;260. WORDS
034054  177777                       -1                      ;FILL WITH -1
```

```
                                        ;*FILL WRITE FROM BUFFER WITH EXPECTED DATA

034056  004037  041350                  JSR     R0,@#FLHEAD     ;SAVE HEADER DATA IN WRFROM
034062  002470                          WRFROM                  ;LOCATION WHERE SAVED
034064  000004                          4                       ;NUMBER OF WORDS SAVED
034066  010000                          10000                   ;FIRST DATA WORD
034070  011025                          <18.*400>!<21.>                 ;SECOND DATA WORD
034072  000000                          0                       ;THIRD DATA WORD
034074  000000                          0                       ;FOURTH DATA WORD
034076  004037  041374                  JSR     R0,@#CLAREA     ;CLEAR 10. WORDS, FROM WRFROM+<4.*2>
034102  002500                          WRFROM+<4.*2>                   ;STARTING FROM WRFROM+<4.*2>
034104  000012                          10.                     ;10. WORDS
034106  001125                          <18.*40>!<21.>                  ;FILL WITH <18.*40>!<21.>

034110  004037  041374                  JSR     R0,@#CLAREA     ;CLEAR 246. WORDS, FROM WRFROM+<14.*2>
034114  002524                          WRFROM+<14.*2>                  ;STARTING FROM WRFROM+<14.*2>
034116  000366                          246.                    ;246. WORDS
034120  177777                          -1                      ;FILL WITH -1


                                        ;*FILL READ HEADER AND DATA COMMAND FOR 10 WORDS

034122  004037  043476                  JSR     R0,@#RUN        ;SETUP TO RUN FOR DATA COMMAND
034126  000000                          0                       ;CYLINDER 0
034130    025                   .BYTE   21.                     ;SECTOR 21.
034131    022                   .BYTE   18.                     ;TRACK 18.
034132  177762                          -10.-4                  ;WORD COUNT (DATA) = 10. +
                                                                ;4 HEADER WORDS
034134  003534                          RE!NTO                  ;BUS ADDRESS
                                                                ;STARTING ADDRESS OF DATA
                                                                ;BUFFER = RE!NTO
034136  000000                          0                       ;DO NOT INHIBIT BUS ADDRESS INCREMENT
034140  014000                          EC!!FMT22                       ;16 BITS PER WORD FORMAT
                                                                ;INHIBIT ECC CORRECTION
                                                                ;DO NOT INHIBIT HEADER COMPARE
034142  002450                          REFOR                   ;GET READY TO DO A REFOR
                                                                ;READ HEADER AND DATA WITH 72 IN RHCS1


034144  004737  041604                  JSR     PC,@#CHECKT     ;CHECK DVA,RDY,MOL,DPR,DRY,VV = 1
                                                                ;AND THAT NO STATUS BITS ARE STUCK = 1
034150  104401  067033                  TYPE    ,CPHALT         ;CANNOT CONTINUE TESTING IF ANY OF
                                                                ;THE FIRST SET OF BITS DON'T = 1
034154  000000                          HALT                    ;STOP

034156  013777  004606  146102          MOV     @#RP4VEC,@RPVEC ;SET RP04 VECTOR ADDRESS
                                                                ;TO 'TIME1' IF P-CLOCK IS PRESENT
                                                                ;OR TO 'TIME2' IF P-CLOCK IS NOT PRESENT
                                                                ;'TIME' WILL ONLY SAVE
                                                                ;CURRENT CYLINDER ADDRESS
                                                                ;AND LOOK AHEAD REGISTERS


034164  013746  002450                  MOV     @#REFOR,-(SP)   ;GET READY TO MOVE COMMAND
034170  052716  000101                  BIS     #GO!IE,(SP)     ;GET READY TO SET 'GO' AND
                                                                ;ENABLE INTERRUPT
```

```
 9119  034174  012677  146100                  MOV     (SP)+,@RHCS1    ;GO WITH
 9120                                                                    ;72 IN RHCS1 FOR READ DATA
 9121                                                                    ;WITH INTERRUPT ENABLED
 9122
 9123
 9124  034200  104413                          WAT                     ;WAIT FOR RDY BIT TO SET
 9125  034202  002300                          RHCS1                   ;WAIT FOR RHCS1 REGISTER
 9126  034204  000200                          RDY                     ;WAIT FOR RDY BIT IN RHCS1 REGISTER
 9127  034206  001614                          908.                    ;ALLOW 9080 MICRO SECONDS
 9128  034210  001507                          839.                    ;RDY MUST SET BETWEEN
 9129                                                                    ;690 AND 17470 MICRO SECONDS
 9130
 9131                                          ;*CHECK READ WORDS
 9132
 9133  034212  004037  043542                  JSR     R0,@#COMPAR     ;COMPARE TWO BLOCKS OF MEMORY
 9134  034216  002470                          WRFROM                  ;GOOD DATA STARTS FROM WRFROM
 9135  034220  003534                          REINTO                  ;TEST DATA STARTS FROM REINTO
 9136  034222  000404                          260.                    ;260. WORDS TO BE COMPARED
 9137  034224  034230                          ST26                    ;RETURN TO ST26 ON ERROR
 9138  034226  034234                          ST27                    ;RETURN TO ST27 ON NO ERROR
 9139
 9140
 9141  034230  104032                  ST26:   ERROR   32              ;READ HEADER AND DATA
 9142  034232  000207                          RTS     PC              ;FOLLOWING A SEEK TO CYLINDER 0
 9143                                                                    ;FROM CYLINDER 10 GAVE AN
 9144                                                                    ;ERROR
 9145
 9146  034234                          ST27:
 9147
```

```
                                ;;*****************************************************************
                                ;*TEST 44       SEEK & READ TEST (CYL = 009)

                                ;*      THIS TEST DEPENDS ON HEADER AND DATA WRITTEN BY THE
                                ;*      PREVIOUS TEST.  AT THIS POINT THE HEADS ARE ON
                                ;*      CYLINDER 0
                                ;*
                                ;*      ALL REGISTERS ARE SAVED
                                ;*      A SEEK IS GIVEN TO CYLINDER 9
                                ;*      ALL REGISTERS ARE CHECKED FOR IMPROPER CHANGE
                                ;*      DATA WRITTEN ON CYLINDER 9 IS CHECKED
                                ;;*****************************************************************
034234  000004                TST44:  SCOPE
034236  012706  001000                MOV     #STACK,SP       ;RESET STACK
034242  012737  000044  004604        MOV     #44,@#TSTNM     ;SAVE TEST NUMBER

034250  004737  041524                JSR     PC,@#CLDISK     ;SET R1-RHCS1  R2-RHCS2
                                                              ;R3-RHDS1, R4-RHER1
                                                              ;GIVE RH-11 INITIALIZE
                                                              ;SETUP UNIT NUMBER

                                ;*THIS GETS HEADS TO CYLINDER 0
034254  013777  004606  146004        MOV     @#RP4VEC,@RPVEC ;SET RP04 VECTOR ADDRESS
                                                              ;TO 'TIME1' IF P-CLOCK IS PRESENT
                                                              ;OR TO 'TIME2' IF P-CLOCK IS NOT PRESENT
                                                              ;'TIME' WILL ONLY SAVE
                                                              ;CURRENT CYLINDER ADDRESS
                                                              ;AND LOOK AHEAD REGISTERS

034262  013746  002426                MOV     @#RECALI,-(SP)  ;GET READY TO MOVE COMMAND
034266  052716  000101                BIS     #GO!IE,(SP)     ;GET READY TO SET 'GO' AND
                                                              ;ENABLE INTERRUPT
034272  012677  146002                MOV     (SP)+,@RHCS1    ;GO WITH
                                                              ;6 IN RHCS1 FOR RECALIBRATE
                                                              ;WITH INTERRUPT ENABLED
034276  011100                        MOV     @R1,R0          ;SAVE RHCS1 DURING ABOVE OPERATION
034300  011305                        MOV     @R3,R5          ;SAVE RHDS1 DURING ABOVE OPERATION

034302  104413                        WAT                     ;WAIT FOR DRY BIT TO SET
034304  002322                        RHDS1                   ;WAIT FOR RHDS1 REGISTER
034306  000200                        DRY                     ;WAIT FOR DRY BIT IN RHDS1 REGISTER
034310  076377                        31999.                  ;ALLOW 319990 MICRO SECONDS
034312  056701                        24001.                  ;DRY MUST SET BETWEEN
                                                              ;79980 AND 560000 MICRO SECONDS

                                ;*FILL REGISTERS FOR A SEEK COMMAND

034314  004037  041474                JSR     R0,@#SEEKCY     ;SEEK FOR
034320  000012                        10.                     ;CYLINDER 10.

                                ;*SAVE REGISTERS FOR COMPARISON AFTER SEEK COMMAND
034322  004037  041672                JSR     R0,@#SAVER      ;SAVE REGISTERS
```

```
9204  034326  002272                          RHWC                    ;RHWC IS THE FIRST REGISTER SAVED
9205  034330  004612                          SAVERE                  ;STARTING ADDRESS OF WHERE
9206                                                                  ;THE REGISTERS ARE SAVED
9207  034332  000022                          18.                     ;NUMBER OF REGISTERS
9208                                                                  ;SAVED = 18.
9209
9210  034334  013777  004606  145724        MOV     @#RP4VEC,@RPVEC ;SET RP04 VECTOR ADDRESS
9211                                                                  ;TO 'TIME1' IF P-CLOCK IS PRESENT
9212                                                                  ;OR TO 'TIME2' IF P-CLOCK IS NOT PRESENT
9213                                                                  ;'TIME' WILL ONLY SAVE
9214                                                                  ;CURRENT CYLINDER ADDRESS
9215                                                                  ;AND LOOK AHEAD REGISTERS
9216
9217
9218  034342  013746  002452                MOV     @#SEECOM,-(SP)  ;GET READY TO MOVE COMMAND
9219  034346  052716  000101                BIS     #GO!IE,(SP)     ;GET READY TO SET 'GO' AND
9220                                                                  ;ENABLE INTERRUPT
9221  034352  012677  145722                MOV     (SP)+,@RHCS1    ;GO WITH
9222                                                                   ;4 IN RHCS1 FOR SEEK
9223                                                                  ;WITH INTERRUPT ENABLED
9224  034356  011100                        MOV     @R1,R0          ;SAVE RHCS1 DURING ABOVE OPERATION
9225  034360  011305                        MOV     @R3,R5          ;SAVE RHDS1 DURING ABOVE OPERATION
9226
9227                                        ;*SEEK FOR ONE CYLINDER=7 MILI SEC., FOR TEN=70 MILI SEC
9228
9229  034362  104413                        WAT                     ;WAIT FOR DRY BIT TO SET
9230  034364  002322                        RHDS1                   ;WAIT FOR RHDS1 REGISTER
9231  034366  000200                        DRY                     ;WAIT FOR DRY BIT IN RHDS1 REGISTER
9232  034370  015530                        7000.                   ;ALLOW 70000 MICRO SECONDS
9233  034372  000043                        35.                     ;DRY MUST SET BETWEEN
9234                                                                  ;69650 AND 70350 MICRO SECONDS
9235
9236                                        ;*COMPARE CONTENTS OF RHCS1 AND RHDS1 ALREADY SAVED IN
9237                                        ;*R0 AND R5 IMMEDIATELY AFTER GO
9238
9239  034374  013746  002452                MOV     @#SEECOM,-(SP)  ,SAVE COMMAND
9240  034400  052716  004301                BIS     #DVA!GO!IE!RDY,(SP)     ;INCLUDE DVA!GO!IE!RDY
9241  034404  011637  001124                MOV     (SP),@#$GDDAT   ;SAVE FOR PRINTOUT
9242  034410  022600                        CMP     (SP)+,R0        ;DURING ABOVE OPERATION ONLY DVA!GO!IE!RDY
9243                                                                  ;AND COMMAND SHOULD BE SET
9244  034412  001405                        BEQ     67$             ;BRANCH IF GOOD
9245  034414  010037  001126                MOV     R0,@#$BDDAT     ;BAD DATA
9246  034420  010137  004600                MOV     R1,@#REGADR     ;FAILING REGISTER RHCS1
9247  034424  104021                        ERROR   21              ;DURING ABOVE OPERATION ONLY
9248                                                                  ;COMMAND AND DVA!GO!IE!RDY SHOULD BE SET
9249  034426  012746  030500        67$:    MOV     #PIP!MOL!DPR!VV,-(SP)   ;SAVE BITS SET DURING OPERATION IN RHDS1
9250  034432  011637  001124                MOV     (SP),@#$GDDAT   ;SAVE FOR PRINTOUT
9251  034436  022605                        CMP     (SP)+,R5        ;DURING ABOVE OPERATION ONLY PIP!MOL!DPR!VV
9252                                                                  ;SHOULD BE SET
9253  034440  001405                        BEQ     69$             ;BRANCH IF GOOD
9254  034442  010537  001126                MOV     R5,@#$BDDAT     ;BAD DATA
9255  034446  010337  004600                MOV     R3,@#REGADR     ;FAILING REGISTER RHDS1
9256  034452  104063                        ERROR   63              ;DURING ABOVE OPERATION ONLY
9257                                                                  ;PIP!MOL!DPR!VV SHOULD BE SET
9258  034454                        69$:
9259
```

```
9260                                        ;*NOW CHANGE SAVED REGISTERS TO EXPECTED VALUE
9261
9262  034454  004037  041426                JSR     R0,@#FILLRE     ;MOV 10. INTO SAVED RHCC
9263  034460  002334                        RHCC                    ;SAVED REGISTER TO CHANGE
9264  034462  000012                        10.                     ;DATA
9265
9266  034464  053737  004740  004636        BIS     @#ATTENT,@#SAVERE+24    ;SET APPROPRIATE 'ATA' BITS
9267                                                                ;FOR WORKING DRIVE IN
9268                                                                ;SAVED RHAS LOACTION
9269
9270  034472  004037  042404                JSR     R0,@#CHREG      ;CHANGE BITS IN SAVED REGISTER
9271  034476  002322                        RHDS1                   ;CHANGE RHDS1 REGISTER
9272
9273  034500  000001                        1                       ;1 BIT/BITS TO BE CHANGED
9274  034502  000001                        1                       ;NEW VALUE OF ATA IS 1
9275  034504  100000                        ATA                     ;CHANGE ATA BIT
9276
9277  034506  004037  042404                JSR     R0,@#CHREG      ;CHANGE BITS IN SAVED REGISTER
9278  034512  002300                        RHCS1                   ;CHANGE RHCS1 REGISTER
9279
9280  034514  000001                        1                       ;1 BIT/BITS TO BE CHANGED
9281  034516  000001                        1                       ;NEW VALUE OF SC IS 1
9282  034520  100000                        SC                      ;CHANGE SC BIT
9283
9284                                        ;*COMPARE REGISTERS AFTER A SEEK COMMAND
9285
9286  034522  004037  042512                JSR     R0,@#COMREG     ;COMPARE SAVED REGISTERS WITH
9287                                                                ;PRESENT VALUE
9288  034526  004612                        SAVERE                  ;GOOD DATA SAVED IN 'SAVERE'
9289  034530  002354                        WC                      ;TEST DATA STARTING FROM 'RHWC'
9290  034532  000022                        18.                     ;18. REGISTERS TO BE COMPARED
9291  034534  034540                        1$                      ;RETURN TO 1$ ON ERROR
9292  034536  034544                        2$                      ;RETURN TO 2$ ON NO ERROR
9293
9294  034540  104037                1$:     ERROR   37              ;SEEK COMMAND CAUSED
9295  034542  000207                        RTS     PC              ;ERROR
9296                                                                ;GOOD DATA GIVES WHAT SHOULD
9297                                                                ;BE THERE
9298                                                                ;RECEIVED DATA GIVES WHAT WAS
9299                                                                ;THERE AFTER A SEEK COMMAND
9300
9301                                        ;*AT THIS POINT THE CURRENT CYLINDER IS GOOD AND THERE ARE
9302                                        ;*NO ERROR BITS
9303                                        ;*A READ HEADER AND DATA WILL BE DONE ON CYLINDER 9
9304                                        ;*SECTOR 21 TRACK 18, EXPECTED DATA IS 23125
9305                                        ;*FOR 20 WORDS
9306                                        ;*CLEAR READ INTO BUFFER WITH ALL ONES
9307
9308  034544                        2$:
9309
9310  034544  004737  041524                JSR     PC,@#CLDISK     ;SET R1-RHCS1, R2-RHCS2
9311                                                                ;R3-RHDS1, R4-RHER1
9312                                                                ;GIVE RH-11 INITIALIZE
9313                                                                ;SETUP UNIT NUMBER
9314  034550  004037  041374                JSR     R0,@#CLAREA     ;CLEAR 260. WORDS, FROM REINTO
9315  034554  003534                        REINTO                  ;STARTING FROM REINTO
```

```
 9316  034556  000404                          260.                    ;260. WORDS
 9317  034560  177777                          -1                      ;FILL WITH -1
 9318
 9319
 9320                                          ;*FILL  WRITE FROM BUFFER WITH EXPECTED DATA
 9321
 9322
 9323  034562  004037  041350                  JSR     R0,@#FLHEAD     ;SAVE HEADER DATA IN WRFROM
 9324  034566  002470                          WRFROM                  ;LOCATION WHERE SAVED
 9325  034570  000004                          4                       ;NUMBER OF WORDS SAVED
 9326  034572  010011                          10011                   ;FIRST DATA WORD
 9327  034574  011025                          <18.*400>!<21.>                 ;SECOND DATA WORD
 9328  034576  000000                          0                       ;THIRD DATA WORD
 9329  034600  000000                          0                       ;FOURTH DATA WORD
 9330  034602  004037  041374                  JSR     R0,@#CLAREA     ;CLEAR 20. WORDS, FROM WRFROM+<4.*2>
 9331  034606  002500                          WRFROM+<4.*2>                   ;STARTING FROM WRFROM+<4.*2>
 9332  034610  000024                          20.                     ;20. WORDS
 9333  034612  023125                          <9.*2000>!<18.*40>!<21.>                        ;FILL WITH <9.*2000>!<18.*40>!<2
 9334
 9335  034614  004037  041374                  JSR     R0,@#CLAREA     ;CLEAR 250. WORDS, FROM WRFROM+<24.*2>
 9336  034620  002550                          WRFROM+<24.*2>                  ;STARTING FROM WRFROM+<24.*2>
 9337  034622  000372                          250.                    ;250. WORDS
 9338  034624  177777                          -1                      ;FILL WITH -1
 9339
 9340
 9341                                          ;*FILL READ HEADER AND DATA COMMAND FOR 10 WORDS
 9342
 9343  034626  004037  043476                  JSR     R0,@#RUN        ;SETUP TO RUN FOR DATA COMMAND
 9344  034632  000011                          9.                      ;CYLINDER 9.
 9345  034634     025                  .BYTE   21.                     ;SECTOR 21.
 9346  034635     022                  .BYTE   18.                     ;TRACK 18.
 9347  034636  177750                          -20.-4                  ;WORD COUNT (DATA) = 20. +
 9348                                                                  ;4 HEADER WORDS
 9349  034640  003534                          REINTO                  ;BUS ADDRESS
 9350                                                                  ;STARTING ADDRESS OF DATA
 9351                                                                  ;BUFFER = REINTO
 9352  034642  000000                          0                       ;DO NOT INHIBIT BUS ADDRESS INCREMENT
 9353  034644  014000                          ECI!FMT22                       ;16 BITS PER WORD FORMAT
 9354                                                                  ;INHIBIT ECC CORRECTION
 9355                                                                  ;DO NOT INHIBIT HEADER COMPARE
 9356  034646  002450                          REFOR                   ;GET READY TO DO A REFOR
 9357                                                                  ;READ HEADER AND DATA WITH 72 IN RHCS1
 9358
 9359
 9360  034650  004737  041604                  JSR     PC,@#CHECKT     ;CHECK DVA,RDY,MOL,DPR,DRY,VV = 1
 9361                                                                  ;AND THAT NO STATUS BITS ARE STUCK = 1
 9362  034654  104401  067033                  TYPE    ,CPHALT         ;CANNOT CONTINUE TESTING IF ANY OF
 9363                                                                  ;THE FIRST SET OF BITS DON'T = 1
 9364  034660  000000                          HALT                    ;STOP
 9365
 9366  034662  013777  004606  145376          MOV     @#RP4VEC,@RPVEC ;SET RP04 VECTOR ADDRESS
 9367                                                                  ;TO 'TIME1' IF P-CLOCK IS PRESENT
 9368                                                                  ;OR TO 'TIME2' IF P-CLOCK IS NOT PRESENT
 9369                                                                  ;'TIME' WILL ONLY SAVE
 9370                                                                  ;CURRENT CYLINDER ADDRESS
 9371                                                                  ;AND LOOK AHEAD REGISTERS
```

```
 9372
 9373
 9374  034670  013746  002450                MOV     @#REFOR,-(SP)   ;GET READY TO MOVE COMMAND
 9375  034674  052716  000101                BIS     #GO!IE,(SP)     ;GET READY TO SET 'GO' AND
 9376                                                                  ;ENABLE INTERRUPT
 9377  034700  012677  145374                MOV     (SP)+,@RHCS1    ;GO WITH
 9378                                                                    ;72 IN RHCS1 FOR READ DATA
 9379                                                                  ;WITH INTERRUPT ENABLED
 9380
 9381
 9382  034704  104413                        WAT                     ;WAIT FOR RDY BIT TO SET
 9383  034706  002300                        RHCS1                   ;WAIT FOR RHCS1 REGISTER
 9384  034710  000200                        RDY                     ;WAIT FOR RDY BIT IN RHCS1 REGISTER
 9385  034712  001614                        908.                    ;ALLOW 9080 MICRO SECONDS
 9386  034714  001507                        839.                    ;RDY MUST SET BETWEEN
 9387                                                                  ;690 AND 17470 MICRO SECONDS
 9388
 9389                                        ;*CHECK READ WORDS
 9390
 9391  034716  004037  043542                JSR     R0,@#COMPAR     ;COMPARE TWO BLOCKS OF MEMORY
 9392  034722  002470                        WRFROM                  ;GOOD DATA STARTS FROM WRFROM
 9393  034724  003534                        REINTO                  ;TEST DATA STARTS FROM REINTO
 9394  034726  000404                        260.                    ;260. WORDS TO BE COMPARED
 9395  034730  034734                        3$                      ;RETURN TO 3$ ON ERROR
 9396  034732  034740                        4$                      ;RETURN TO 4$ ON NO ERROR
 9397
 9398
 9399  034734  104032              3$:       ERROR   32              ;READ HEADER AND DATA
 9400  034736  000207                        RTS     PC              ;FOLLOWING A SEEK TO CYLINDER 9
 9401                                                                  ;FROM CYLINDER 0 GAVE AN
 9402                                                                  ;ERROR
 9403  034740                    4$:
 9404
 9405
 9406
 9407
```

```
 9408                                    .SBTTL  WRITE CHECK DATA & WRITE PROTECT TESTS
 9409
```

```
                                ;;*****************************************************************
                                ;*TEST 45       WRITE CHECK HEADER AND DATA
                                ;*
                                ;*      WRITE CHECK HEADER AND DATA CYLINDER 5 FORMAT 16 BITS PER WORD
                                ;*      TRACK 7, SECTOR 4, KEYS 0, NUMBER OF WORDS 266
                                ;*      CONSISTING OF
                                ;*      10 WORDS OF 12344 (6 BITS FOR CYL, 5 FOR TRACK, 5 FOR SECTOR)
                                ;*      10 WORDS OF 177777
                                ;*      10 WORDS OF 0
                                ;*      10 WORDS OF 052525
                                ;*      10 WORDS OF 125252
                                ;*      16 WORDS OF LEFT ROTATING ZERO (EG 177776,177775)
                                ;*      16 WORDS OF LEFT ROTATING ONE (EG 1,2,4,10)
                                ;*      174 WORDS OF 377
                                ;*      4 WORDS OF HEADER
                                ;*      2 WORDS OF 12345
                                ;*
                                ;*
                                ;*      FIRST THE ABOVE DATA IS WRITTEN BY A WRITE HEADER AND
                                ;*      DATA COMMAND
                                ;*      CHECK FOR NO ERRORS
                                ;*      THEN THE ABOVE DATA IS READ BY A READ HEADER AND DATA
                                ;*      CHECK FOR NO ERRORS
                                ;*      THEN THE ABOVE WRITE CHECK HEADER AND DATA IS GIVEN
                                ;;*****************************************************************
034740  000004                  TST45:  SCOPE
034742  012706  001000                  MOV     #STACK,SP       ;RESET STACK
034746  012737  000045  004604          MOV     #45,@#TSTNM     ;SAVE TEST NUMBER
034754  004737  041524                  JSR     PC,@#CLDISK     ;SET R1-RHCS1, R2-RHCS2
                                                                ;R3-RHDS1, R4-RHER1
                                                                ;GIVE RH-11 INITIALIZE
                                                                ;SETUP UNIT NUMBER
034760  004737  041604                  JSR     PC,@#CHECKT     ;CHECK DVA,RDY,MOL,DPR,DRY,VV = 1
                                                                ;AND THAT NO STATUS BITS ARE STUCK = 1
034764  104401  067033                  TYPE    ,CPHALT         ;CANNOT CONTINUE TESTING IF ANY OF
                                                                ;THE FIRST SET OF BITS DON'T = 1
034770  000000                          HALT                    ;STOP
                                ;*GET HEADS TO CYLINDER 5
034772  004037  041474                  JSR     R0,@#SEEKCY     ;SEEK FOR
034776  000005                  5                               ;CYLINDER 5
035000  013777  004606  145260          MOV     @#RP4VEC,@RPVEC ;SET RP04 VECTOR ADDRESS
                                                                ;TO 'TIME1' IF P-CLOCK IS PRESENT
                                                                ;OR TO 'TIME2' IF P-CLOCK IS NOT PRESENT
                                                                ;'TIME' WILL ONLY SAVE
                                                                ;CURRENT CYLINDER ADDRESS
                                                                ;AND LOOK AHEAD REGISTERS
035006  013746  002452                  MOV     @#SEECOM,-(SP)  ;GET READY TO MOVE COMMAND
035012  052716  000101                  BIS     #GO!IE,(SP)     ;GET READY TO SET 'GO' AND
                                                                ;ENABLE INTERRUPT
035016  012677  145256                  MOV     (SP)+,@RHCS1    ;GO WITH
```

```
                                                        ;4 IN RHCS1 FOR SEEK
                                                        ;WITH INTERRUPT ENABLED
035022  011100                  MOV     @R1,R0          ;SAVE RHCS1 DURING ABOVE OPERATION
035024  011305                  MOV     @R3,R5          ;SAVE RHDS1 DURING ABOVE OPERATION

035026  104413                  WAT                     ;WAIT FOR DRY BIT TO SET
035030  002322                  RHDS1                   ;WAIT FOR RHDS1 REGISTER
035032  000200                  DRY                     ;WAIT FOR DRY BIT IN RHDS1 REGISTER
035034  015530                  7000.                   ;ALLOW 70000 MICRO SECONDS
035036  000043                  35.                     ;DRY MUST SET BETWEEN
                                                        ;69650 AND 70350 MICRO SECONDS

035040  004737  041524          JSR     PC,@#CLDISK     ;SET R1-RHCS1, R2-RHCS2
                                                        ;R3-RHDS1, R4-RHER1
                                                        ;GIVE RH-11 INITIALIZE
                                                        ;SETUP UNIT NUMBER

                                ;*FILL WRITE FROM BUFFER WITH HEADER

035044  004037  041350          JSR     R0,@#FLHEAD     ;SAVE HEADER DATA IN WRFROM
035050  002470                  WRFROM                  ;LOCATION WHERE SAVED
035052  000004                  4                       ;NUMBER OF WORDS SAVED
035054  010005                  10005                   ;FIRST DATA WORD
035056  003404                  <7*400>!4                       ;SECOND DATA WORD
035060  000000                  0                       ;THIRD DATA WORD
035062  000000                  0                       ;FOURTH DATA WORD

                                ;*10 WORDS OF OF THE FOLLOWING DATA
                                ;* 12344,17777,0,52525,125252

035064  004037  041374          JSR     R0,@#CLAREA     ;CLEAR 10. WORDS, FROM WRFROM+<4*2>
035070  002500                  WRFROM+<4*2>                    ;STARTING FROM WRFROM+<4*2>
035072  000012                  10.                     ;10. WORDS
035074  012344                  <5*2000>!<7*40>!4                       ;FILL WITH <5*2000>!<7*40>!4

035076  004037  041374          JSR     R0,@#CLAREA     ;CLEAR 10. WORDS, FROM WRFROM+<14.*2>
035102  002524                  WRFROM+<14.*2>                  ;STARTING FROM WRFROM+<14.*2>
035104  000012                  10.                     ;10. WORDS
035106  177777                  -1                      ;FILL WITH -1

035110  004037  041374          JSR     R0,@#CLAREA     ;CLEAR 10. WORDS, FROM WRFROM+<24.*2>
035114  002550                  WRFROM+<24.*2>                  ;STARTING FROM WRFROM+<24.*2>
035116  000012                  10.                     ;10. WORDS
035120  000000                  0                       ;FILL WITH 0

035122  004037  041374          JSR     R0,@#CLAREA     ;CLEAR 10. WORDS, FROM WRFROM+<34.*2>
035126  002574                  WRFROM+<34.*2>                  ;STARTING FROM WRFROM+<34.*2>
035130  000012                  10.                     ;10. WORDS
035132  052525                  52525                   ;FILL WITH 52525

035134  004037  041374          JSR     R0,@#CLAREA     ;CLEAR 10. WORDS, FROM WRFROM+<44.*2>
035140  002620                  WRFROM+<44.*2>                  ;STARTING FROM WRFROM+<44.*2>
035142  000012                  10.                     ;10. WORDS
035144  125252                  125252                  ;FILL WITH 125252
```

```
                                        ;*FILL LEFT ROTATING ZEROS FROM WRFROM+<54.*2>
                                        MOV     R1,-(SP)        ;;PUSH R1 ON STACK
                                        MOV     #177776,R0      ;DATA
                                        MOV     #16.,R5         ;COUNT
                                        MOV     #WRFROM+<54.*2>,R1 ;WHERE DATA GOES
                                        SEC
                                1$:     MOV     R0,(R1)+        ;STORE DATA
                                        ROL     R0              ;GET ZERO ONE BIT LEFT
                                        DEC     R5              ;COUNT 16
                                        BNE     1$              ;BRANCH IF 16 NOT DONE

                                        ;*FILL LEFT ROTATING ONE INTO WRFROM+<65.*2>

                                        CLC
                                        MOV     #1,R0
                                2$:     MOV     R0,(R1)+
                                        ASL     R0
                                        BCC     2$
                                        MOV     (SP)+,R1        ;;POP STACK INTO R1


                                        ;*FILL REST OF DATA

                                        JSR     R0,@#CLAREA     ;CLEAR 174. WORDS, FROM WRFROM+<86.*2>
                                        WRFROM+<86.*2>                  ;STARTING FROM WRFROM+<86.*2>
                                        174.                    ;174. WORDS
                                        377                     ;FILL WITH 377


                                        JSR     R0,@#FLHEAD     ;SAVE HEADER DATA IN WRFROM+<260.*2>
                                        WRFROM+<260.*2>                 ;LOCATION WHERE SAVED
                                        4                       ;NUMBER OF WORDS SAVED
                                        10005                   ;FIRST DATA WORD
                                        <7*400>!5                       ;SECOND DATA WORD
                                        0                       ;THIRD DATA WORD
                                        0                       ;FOURTH DATA WORD
                                        JSR     R0,@#CLAREA     ;CLEAR 2 WORDS, FROM WRFROM+<264.*2>
                                        WRFROM+<264.*2>                 ;STARTING FROM WRFROM+<264.*2>
                                        2                       ;2 WORDS
                                        <5*2000>!<7*40>!5                       ;FILL WITH <5*2000>!<7*40>!5


                                        ;*READ INTO BUFFER WILL BE CLEARED
                                        JSR     R0,@#CLAREA     ;CLEAR 266. WORDS, FROM REINTO
                                        REINTO                  ;STARTING FROM REINTO
                                        266.                    ;266. WORDS
                                        177400                  ;FILL WITH 177400


                                        ;*THE WRITE HEADER AND DATA COMMAND WILL BE LOADED

                                        JSR     R0,@#RUN        ;SETUP TO RUN FOR DATA COMMAND
                                        5                       ;CYLINDER 5
```

```
9578  035300  004                  .BYTE   4                       ;SECTOR 4
9579  035301  007                  .BYTE   7                       ;TRACK 7
9580  035302  177366               -262.-4                         ;WORD COUNT (DATA) = 262. +
9581                                                                 ;4 HEADER WORDS
9582  035304  002470               WRFROM                          ;BUS ADDRESS
9583                                                                 ;STARTING ADDRESS OF DATA
9584                                                                 ;BUFFER = WRFROM
9585  035306  000000               0                               ;DO NOT INHIBIT BUS ADDRESS INCREMENT
9586  035310  010000               FMT22                           ;16 BITS PER WORD FORMAT
9587                                                                 ;DO NOT INHIBIT ECC CORRECTION
9588                                                                 ;DO NOT INHIBIT HEADER COMPARE
9589  035312  002444               WRIFOR                          ;GET READY TO DO A WRIFOR
9590                                                       ;WRITE HEADER AND DATA WITH 62 IN RHCS1
9591
9592
9593                               ;*NOW SAVE REGISTERS FOR COMPARISON AFTER WRITE HEADER AND DATA
9594
9595  035314  004037  041672       JSR     R0,@#SAVER              ;SAVE REGISTERS
9596  035320  002272               RHWC                            ;RHWC IS THE FIRST REGISTER SAVED
9597  035322  004612               SAVERE                          ;STARTING ADDRESS OF WHERE
9598                                                                 ;THE REGISTERS ARE SAVED
9599  035324  000022               18.                             ;NUMBER OF REGISTERS
9600                                                                 ;SAVED = 18.
9601
9602  035326  004737  041604       JSR     PC,@#CHECKT             ;CHECK DVA,RDY,MOL,DPR,DRY,VV = 1
9603                                                                 ;AND THAT NO STATUS BITS ARE STUCK = 1
9604  035332  104401  067033       TYPE    ,CPHALT                 ;CANNOT CONTINUE TESTING IF ANY OF
9605                                                                 ;THE FIRST SET OF BITS DON'T = 1
9606  035336  000000               HALT                            ;STOP
9607
9608  035340  013777  004606  144720  MOV  @#RP4VEC,@RPVEC         ;SET RP04 VECTOR ADDRESS
9609                                                                 ;TO 'TIME1' IF P-CLOCK IS PRESENT
9610                                                                 ;OR TO 'TIME2' IF P-CLOCK IS NOT PRESENT
9611                                                                 ;'TIME' WILL ONLY SAVE
9612                                                                 ;CURRENT CYLINDER ADDRESS
9613                                                                 ;AND LOOK AHEAD REGISTERS
9614
9615
9616  035346  013746  002444       MOV     @#WRIFOR,-(SP)          ;GET READY TO MOVE COMMAND
9617  035352  052716  000101       BIS     #GO!IE,(SP)             ;GET READY TO SET 'GO' AND
9618                                                                 ;ENABLE INTERRUPT
9619  035356  012677  144716       MOV     (SP)+,@RHCS1            ;GO WITH
9620                                                                  ;62 IN RHCS1 FOR WRITE HEADER AND DATA
9621                                                                 ;WITH INTERRUPT ENABLED
9622
9623
9624  035362  104413               WAT                             ;WAIT FOR RDY BIT TO SET
9625  035364  002300               RHCS1                           ;WAIT FOR RHCS1 REGISTER
9626  035366  000200               RDY                             ;WAIT FOR RDY BIT IN RHCS1 REGISTER
9627  035370  001732               986.                            ;ALLOW 9860 MICRO SECONDS
9628  035372  001502               834.                            ;RDY MUST SET BETWEEN
9629                                                                 ;1520 AND 18200 MICRO SECONDS
9630
9631                               ;*NOW CHANGE SAVED REGISTERS TO EXPECTED VALUES
9632  035374  004037  041426       JSR     R0,@#FILLRE             ;MOV 0 INTO SAVED RHWC
9633  035400  002272               RHWC                            ;SAVED REGISTER TO CHANGE
```

```
 9634  035402  000000                          0                       ;DATA
 9635  035404  004037  041426                  JSR     R0,@#FILLRE     ;MOV WRFROM+<266.*2> INTO SAVED RHBA
 9636  035410  002274                          RHBA                    ;SAVED REGISTER TO CHANGE
 9637  035412  003514                          WRFROM+<266.*2>                 ;DATA
 9638  035414  004037  041426                  JSR     R0,@#FILLRE     ;MOV 3406 INTO SAVED RHDST
 9639  035420  002304                          RHDST                   ;SAVED REGISTER TO CHANGE
 9640  035422  003406                          3406                    ;DATA
 9641
 9642                                          ;*NOW COMPARE REGISTERS BEFORE WRITE HEADER AND DATA
 9643                                          ;*WITH REGISTERS AFTER COMMAND
 9644
 9645
 9646  035424  004037  042512                  JSR     R0,@#COMREG     ;COMPARE SAVED REGISTERS WITH
 9647                                                                  ;PRESENT VALUE
 9648  035430  004612                          SAVERE                  ;GOOD DATA SAVED IN 'SAVERE'
 9649  035432  002354                          WC                      ;TEST DATA STARTING FROM 'RHWC'
 9650  035434  000022                          18.                     ;18. REGISTERS TO BE COMPARED
 9651  035436  035442                          3$                      ;RETURN TO 3$ ON ERROR
 9652  035440  035446                          4$                      ;RETURN TO 4$ ON NO ERROR
 9653
 9654  035442  104027                3$:       ERROR   27              ;WRITE HEADER AND DATA
 9655  035444  000207                          RTS     PC              ;CAUSED IMPROPER REGISTER
 9656                                                                  ;CHANGE
 9657                                                                  ;GOOD DATA GIVES WHAT SHOULD
 9658                                                                  ;BE THERE
 9659                                                                  ;RECEIVED DATA GIVES WHAT
 9660                                                                  ;WAS THERE AFTER COMMANT
 9661
 9662                                          ;*NOW FILL COMMAND FOR READ
 9663
 9664  035446                        4$:
 9665
 9666  035446  004737  041524                  JSR     PC,@#CLDISK     ;SET R1-RHCS1, R2-RHCS2
 9667                                                                  ;R3-RHDS1, R4-RHER1
 9668                                                                  ;GIVE RH-11 INITIALIZE
 9669                                                                  ;SETUP UNIT NUMBER
 9670
 9671  035452  004037  043476                  JSR     R0,@#RUN        ;SETUP TO RUN FOR DATA COMMAND
 9672  035456  000005                          5                       ;CYLINDER 5
 9673  035460     004                .BYTE     4                       ;SECTOR 4
 9674  035461     007                .BYTE     7                       ;TRACK 7
 9675  035462  177366                          -262.-4                 ;WORD COUNT (DATA) = 262. +
 9676                                                                  ;4 HEADER WORDS
 9677  035464  003534                          REINTO                  ;BUS ADDRESS
 9678                                                                  ;STARTING ADDRESS OF DATA
 9679                                                                  ;BUFFER = REINTO
 9680  035466  000000                          0                       ;DO NOT INHIBIT BUS ADDRESS INCREMENT
 9681  035470  014000                          EC1!FMT22                       ;16 BITS PER WORD FORMAT
 9682                                                                  ;INHIBIT ECC CORRECTION
 9683                                                                  ;DO NOT INHIBIT HEADER COMPARE
 9684  035472  002450                          REFOR                   ;GET READY TO DO A REFOR
 9685                                                                  ;READ HEADER AND DATA WITH 72 IN RHCS1
 9686
 9687
 9688                                          ;*NOW SAVE REGISTERS FOR COMPARISON AFTER READ HEADER AND DATA
 9689
```

```
 9690  035474  004037  041672                JSR     R0,@#SAVER      ;SAVE REGISTERS
 9691  035500  002272                        RHWC                    ;RHWC IS THE FIRST REGISTER SAVED
 9692  035502  004612                        SAVERE                  ;STARTING ADDRESS OF WHERE
 9693                                                                ;THE REGISTERS ARE SAVED
 9694  035504  000022                        18.                     ;NUMBER OF REGISTERS
 9695                                                                ;SAVED = 18.
 9696
 9697  035506  004737  041604                JSR     PC,@#CHECKT     ;CHECK DVA,RDY,MOL,DPR,DRY,VV = 1
 9698                                                                ;AND THAT NO STATUS BITS ARE STUCK = 1
 9699  035512  104401  067033                TYPE    ,CPHALT         ;CANNOT CONTINUE TESTING IF ANY OF
 9700                                                                ;THE FIRST SET OF BITS DON'T = 1
 9701  035516  000000                        HALT                    ;STOP
 9702
 9703  035520  013777  004606  144540        MOV     @#RP4VEC,@RPVEC ;SET RP04 VECTOR ADDRESS
 9704                                                                ;TO 'TIME1' IF P-CLOCK IS PRESENT
 9705                                                                ;OR TO 'TIME2' IF P-CLOCK IS NOT PRESENT
 9706                                                                ;'TIME' WILL ONLY SAVE
 9707                                                                ;CURRENT CYLINDER ADDRESS
 9708                                                                ;AND LOOK AHEAD REGISTERS
 9709
 9710
 9711  035526  013746  002450                MOV     @#REFOR,-(SP)   ;GET READY TO MOVE COMMAND
 9712  035532  052716  000101                BIS     #GO!IE,(SP)     ;GET READY TO SET 'GO' AND
 9713                                                                ;ENABLE INTERRUPT
 9714  035536  012677  144536                MOV     (SP)+,@RHCS1    ;GO WITH
 9715                                                                  ;72 IN RHCS1 FOR READ DATA
 9716                                                                ;WITH INTERRUPT ENABLED
 9717
 9718
 9719  035542  104413                        WAT                     ;WAIT FOR RDY BIT TO SET
 9720  035544  002300                        RHCS1                   ;WAIT FOR RHCS1 REGISTER
 9721  035546  000200                        RDY                     ;WAIT FOR RDY BIT IN RHCS1 REGISTER
 9722  035550  001732                        986.                    ;ALLOW 9860 MICRO SECONDS
 9723  035552  001502                        834.                    ;RDY MUST SET BETWEEN
 9724                                                                ;1520 AND 18200 MICRO SECONDS
 9725
 9726                                        ;*CHANGE SAVED REGISTERS TO EXPECTED VALUES
 9727
 9728  035554  004037  041426                JSR     R0,@#FILLRE     ;MOV 0 INTO SAVED RHWC
 9729  035560  002272                        RHWC                    ;SAVED REGISTER TO CHANGE
 9730  035562  000000                        0                       ;DATA
 9731  035564  004037  041426                JSR     R0,@#FILLRE     ;MOV REINTO+<266.*2> INTO SAVED RHBA
 9732  035570  002274                        RHBA                    ;SAVED REGISTER TO CHANGE
 9733  035572  004560                        REINTO+<266.*2>                 ;DATA
 9734  035574  004037  041426                JSR     R0,@#FILLRE     ;MOV 3406 INTO SAVED RHDST
 9735  035600  002304                        RHDST                   ;SAVED REGISTER TO CHANGE
 9736  035602  003406                        3406                    ;DATA
 9737
 9738                                        ;*COMPARE REGISTERS BEFORE READ HEADER AND DATA
 9739                                        ;*WITH REGISTERS AFTER COMMAND
 9740
 9741
 9742
 9743  035604  004037  042512                JSR     R0,@#COMREG     ;COMPARE SAVED REGISTERS WITH
 9744                                                                ;PRESENT VALUE
 9745  035610  004612                        SAVERE                  ;GOOD DATA SAVED IN 'SAVERE'
```

```
 9746  035612  002354                          WC                      ;TEST DATA STARTING FROM 'RHWC'
 9747  035614  000022                          18.                     ;18. REGISTERS TO BE COMPARED
 9748  035616  035622                          5$                      ;RETURN TO 5$ ON ERROR
 9749  035620  035626                          6$                      ;RETURN TO 6$ ON NO ERROR
 9750
 9751
 9752  035622  104031                  5$:     ERROR   31              ;READ HEADER AND DATA CAUSED
 9753  035624  000207                          RTS     PC              ;IMPROPER REGISTER CHANGE
 9754                                                                  ;GOOD DATA GIVES WHAT SHOULD
 9755                                                                  ;BE THERE RECEIVED DATA GIVES WHAT WAS
 9756                                                                  ;RECEIVED DATA GIVES WHAT WAS
 9757                                                                  ;THERE AFTER COMMAND
 9758
 9759                                  ;*NOW READ INTO BUFFER WILL BE CHECKED TO SEE
 9760                                  ;*THAT READ WAS GOOD
 9761
 9762  035626                  6$:
 9763
 9764  035626  004037  043542                  JSR     R0,@#COMPAR     ;COMPARE TWO BLOCKS OF MEMORY
 9765  035632  002470                          WRFROM                  ;GOOD DATA STARTS FROM WRFROM
 9766  035634  003534                          REINTO                  ;TEST DATA STARTS FROM REINTO
 9767  035636  000412                          266.                    ;266. WORDS TO BE COMPARED
 9768  035640  035644                          7$                      ;RETURN TO 7$ ON ERROR
 9769  035642  035650                          10$                     ;RETURN TO 10$ ON NO ERROR
 9770
 9771
 9772
 9773  035644  104032                  7$:     ERROR   32              ;WRITE HEADER AND DATA
 9774  035646  000207                          RTS     PC              ;FOLLOWED BY A READ HEADER
 9775                                                                  ;AND DATA GAVE A READ ERROR
 9776                                                                  ;ERROR MAY BE IN READ OR WRITE
 9777
 9778                                  ;*A WRITE,READ HAS BEEN SUCCESSFULLY COMPLETED
 9779                                  ;*NOW A WRITE CHECK HEADER AND DATA WILL BE GIVEN
 9780                                  ;*FILL THE WRITE CHECK HEADER AND DATA
 9781
 9782  035650                  10$:
 9783
 9784  035650  004737  041524                  JSR     PC,@#CLDISK     ;SET R1-RHCS1, R2-RHCS2
 9785                                                                  ;R3-RHDS1, R4-RHER1
 9786                                                                  ;GIVE RH-11 INITIALIZE
 9787                                                                  ;SETUP UNIT NUMBER
 9788
 9789  035654  004037  043476                  JSR     R0,@#RUN        ;SETUP TO RUN FOR DATA COMMAND
 9790  035660  000005                          5                       ;CYLINDER 5
 9791  035662     004          .BYTE   4       ;SECTOR 4
 9792  035663     007          .BYTE   7       ;TRACK 7
 9793  035664  177366                          -262.-4                 ;WORD COUNT (DATA) = 262. +
 9794                                                                  ;4 HEADER WORDS
 9795  035666  002470                          WRFROM                  ;BUS ADDRESS
 9796                                                                  ;STARTING ADDRESS OF DATA
 9797                                                                  ;BUFFER = WRFROM
 9798  035670  000000                          0                       ;DO NOT INHIBIT BUS ADDRESS INCREMENT
 9799  035672  014000                          EC1!FMT22                       ;16 BITS PER WORD FORMAT
 9800                                                                  ;INHIBIT ECC CORRECTION
 9801                                                                  ;DO NOT INHIBIT HEADER COMPARE
```

```
 9802  035674  002440                          WRCHDT                  ;GET READY TO DO A WRCHDT
 9803                                                                  ;WRITE CHECK HEADER AND DATA WITH 52 IN RHCS1
 9804
 9805
 9806                                          ;*SAVE REGISTERS FOR COMPARISON AFTER WRITE CHECK
 9807
 9808  035676                          ST25:
 9809  035676  004037  041672                  JSR     R0,@#SAVER      ;SAVE REGISTERS
 9810  035702  002272                          RHWC                    ;RHWC IS THE FIRST REGISTER SAVED
 9811  035704  004612                          SAVERE                  ;STARTING ADDRESS OF WHERE
 9812                                                                  ;THE REGISTERS ARE SAVED
 9813  035706  000022                          18.                     ;NUMBER OF REGISTERS
 9814                                                                  ;SAVED = 18.
 9815
 9816  035710  004737  041604                  JSR     PC,@#CHECKT     ;CHECK DVA,RDY,MOL,DPR,DRY,VV = 1
 9817                                                                  ;AND THAT NO STATUS BITS ARE STUCK = 1
 9818  035714  104401  067033                  TYPE    ,CPHALT         ;CANNOT CONTINUE TESTING IF ANY OF
 9819                                                                  ;THE FIRST SET OF BITS DON'T = 1
 9820  035720  000000                          HALT                    ;STOP
 9821
 9822  035722  013777  004606  144336          MOV     @#RP4VEC,@RPVEC ;SET RP04 VECTOR ADDRESS
 9823                                                                  ;TO 'TIME1' IF P-CLOCK IS PRESENT
 9824                                                                  ;OR TO 'TIME2' IF P-CLOCK IS NOT PRESENT
 9825                                                                  ;'TIME' WILL ONLY SAVE
 9826                                                                  ;CURRENT CYLINDER ADDRESS
 9827                                                                  ;AND LOOK AHEAD REGISTERS
 9828
 9829
 9830  035730  013746  002440          MOV     @#WRCHDT,-(SP)  ;GET READY TO MOVE COMMAND
 9831  035734  052716  000101          BIS     #GO!IE,(SP)     ;GET READY TO SET 'GO' AND
 9832                                                           ;ENABLE INTERRUPT
 9833  035740  012677  144334          MOV     (SP)+,@RHCS1    ;GO WITH
 9834                                                            ;52 IN RHCS1 FOR WRITE CHECK HEADER AND DATA
 9835                                                           ;WITH INTERRUPT ENABLED
 9836  035744  011100                  MOV     @R1,R0          ;SAVE RHCS1 DURING ABOVE OPERATION
 9837  035746  011305                  MOV     @R3,R5          ;SAVE RHDS1 DURING ABOVE OPERATION
 9838
 9839
 9840  035750  104413                  WAT                     ;WAIT FOR RDY BIT TO SET
 9841  035752  002300                  RHCS1                   ;WAIT FOR RHCS1 REGISTER
 9842  035754  000200                  RDY                     ;WAIT FOR RDY BIT IN RHCS1 REGISTER
 9843  035756  001732                  986.                    ;ALLOW 9860 MICRO SECONDS
 9844  035760  001502                  834.                    ;RDY MUST SET BETWEEN
 9845                                                           ;1520 AND 18200 MICRO SECONDS
 9846
 9847                                  ;*COMPARE CONTENTS OF RHCS1 AND RHDS1 ALREADY SAVED IN
 9848                                  ;*R0 AND R5 IMMEDIATELY AFTER GO
 9849
 9850  035762  013746  002440          MOV     @#WRCHDT,-(SP)  ;SAVE COMMAND
 9851  035766  052716  004101          BIS     #IE!DVA!GO,(SP) ;INCLUDE IE!DVA!GO
 9852  035772  011637  001124          MOV     (SP),@#$GDDAT   ;SAVE FOR PRINTOUT
 9853  035776  022600                  CMP     (SP)+,R0        ;DURING ABOVE OPERATION ONLY IE!DVA!GO
 9854                                                           ;AND COMMAND SHOULD BE SET
 9855  036000  001405                  BEQ     64$             ;BRANCH IF GOOD
 9856  036002  010037  001126          MOV     R0,@#$BDDAT     ;BAD DATA
 9857  036006  010137  004600          MOV     R1,@#REGADR     ;FAILING REGISTER RHCS1
```

```
 9858  036012  104021                          ERROR   21              ;DURING ABOVE OPERATION ONLY
 9859                                                                  ;COMMAND AND IE!DVA!GO SHOULD BE SET
 9860  036014  012746  010500          64$:    MOV     #MOL!DPR!VV,-(SP)       ;SAVE BITS SET DURING OPERATION IN RHDS1
 9861  036020  011637  001124                  MOV     (SP),@#$GDDAT   ;SAVE FOR PRINTOUT
 9862  036024  022605                          CMP     (SP)+,R5        ;DURING ABOVE OPERATION ONLY MOL!DPR!VV
 9863                                                                  ;SHOULD BE SET
 9864  036026  001405                          BEQ     66$             ;BRANCH IF GOOD
 9865  036030  010537  001126                  MOV     R5,@#$BDDAT     ;BAD DATA
 9866  036034  010337  004600                  MOV     R3,@#REGADR     ;FAILING REGISTER RHDS1
 9867  036040  104063                          ERROR   63              ;DURING ABOVE OPERATION ONLY
 9868                                                                  ;MOL!DPR!VV SHOULD BE SET
 9869  036042                          66$:
 9870
 9871                                          ;*CHANGE SAVED REGISTERS TO EXPECTED VALUES
 9872
 9873  036042  004037  041426                  JSR     R0,@#FILLRE     ;MOV 0 INTO SAVED RHWC
 9874  036046  002272                          RHWC                    ;SAVED REGISTER TO CHANGE
 9875  036050  000000                          0                       ;DATA
 9876  036052  004037  041426                  JSR     R0,@#FILLRE     ;MOV WRFROM+<266.*2> INTO SAVED RHBA
 9877  036056  002274                          RHBA                    ;SAVED REGISTER TO CHANGE
 9878  036060  003514                          WRFROM+<266.*2>                 ;DATA
 9879  036062  004037  041426                  JSR     R0,@#FILLRE     ;MOV 3406 INTO SAVED RHDST
 9880  036066  002304                          RHDST                   ;SAVED REGISTER TO CHANGE
 9881  036070  003406                          3406                    ;DATA
 9882
 9883                                          ;*COMPARE REGISTERS BEFORE WRITE CHECK HEADER AND DATA
 9884                                          ;*WITH REGISTERS AFTER COMMAND
 9885
 9886
 9887  036072  004037  042512                  JSR     R0,@#COMREG     ;COMPARE SAVED REGISTERS WITH
 9888                                                                  ;PRESENT VALUE
 9889  036076  004612                          SAVERE                  ;GOOD DATA SAVED IN 'SAVERE'
 9890  036100  002354                          WC                      ;TEST DATA STARTING FROM 'RHWC'
 9891  036102  000022                          18.                     ;18. REGISTERS TO BE COMPARED
 9892  036104  036110                          8$                      ;RETURN TO 8$ ON ERROR
 9893  036106  036114                          9$                      ;RETURN TO 9$ ON NO ERROR
 9894
 9895  036110  104040          8$:     ERROR   40              ;WRITE CHECK CAUSED
 9896  036112  000207                  RTS     PC              ;AN IMPROPER REGISTER
 9897                                                          ;CHANGE
 9898                                                          ;GOOD DATA GIVES WHAT
 9899                                                          ;SHOULD BE THERE
 9900                                                          ;RECEIVED DATA GIVES WHAT
 9901                                                          ;WAS THERE AFTER COMMAND
 9902
 9903
 9904  036114                          9$:
```

```
                                ;;*****************************************************************
                                ;*TEST 46       WRITE CHECK DATA
                                ;*      THE DATA FOR THIS TEST IS WRITTEN ON DISK BY PREVIOUS TEST
                                ;*      WRITE CHECK DATA CYLINDER 5, FORMAT 16 BITS PER WORDS
                                ;*      TRACK 7, SECTOR 4, KEYS 0, NUMBER OF WORDS 258
                                ;*      CONSISTING OF
                                ;*      10 WORDS OF 12344 (6 BITS FOR CYL, 5 FOR TRACK, 5 FOR SECTOR)
                                ;*      10 WORDS OF 177777
                                ;*      10 WORDS OF 0
                                ;*      10 WORDS OF 052525
                                ;*      10 WORD OF 125252
                                ;*      16 WORDS OF LEFT ROTATING ZERO (EG177776,177775)
                                ;*      16 WORDS OF LEFT ROTATING ONE (EG 1,2,4,10)
                                ;*      174 WORDS OF 377
                                ;*      2 WORDS OF 12345
                                ;*
                                ;*
                                ;*      FIRST THE ABOVE DATA IS FILLED INTO WRITE FROM BUFFER
                                ;*      THEN THE ABOVE WRITE CHECK DATA IS GIVEN
                                ;;*****************************************************************
036114  000004                  TST46:  SCOPE
036116  012706  001000                  MOV     #STACK,SP       ;RESET STACK
036122  012737  000046  004604          MOV     #46,@#TSTNM     ;SAVE TEST NUMBER
036130  004737  041524                  JSR     PC,@#CLDISK     ;SET R1-RHCS1, R2-RHCS2
                                                                ;R3-RHDS1, R4-RHER1
                                                                ;GIVE RH-11 INITIALIZE
                                                                ;SETUP UNIT NUMBER
036134  004737  041604          JSR     PC,@#CHECKT     ;CHECK DVA,RDY,MOL,DPR,DRY,VV = 1
                                                        ;AND THAT NO STATUS BITS ARE STUCK = 1
036140  104401  067033          TYPE    ,CPHALT         ;CANNOT CONTINUE TESTING IF ANY OF
                                                        ;THE FIRST SET OF BITS DON'T = 1
036144  000000                  HALT                    ;STOP
                                ;*GET HEADS TO CYLINDER 5
036146  004037  041474          JSR     R0,@#SEEKCY     ;SEEK FOR
036152  000005                  5                       ;CYLINDER 5
036154  013777  004606  144104  MOV     @#RP4VEC,@RPVEC ;SET RP04 VECTOR ADDRESS
                                                        ;TO 'TIME1' IF P-CLOCK IS PRESENT
                                                        ;OR TO 'TIME2' IF P-CLOCK IS NOT PRESENT
                                                        ;'TIME' WILL ONLY SAVE
                                                        ;CURRENT CYLINDER ADDRESS
                                                        ;AND LOOK AHEAD REGISTERS
036162  013746  002452          MOV     @#SEECOM,-(SP)  ;GET READY TO MOVE COMMAND
036166  052716  000101          BIS     #GO!IE,(SP)     ;GET READY TO SET 'GO' AND
                                                        ;ENABLE INTERRUPT
036172  012677  144102          MOV     (SP)+,@RHCS1    ;GO WITH
                                                         ;4 IN RHCS1 FOR SEEK
                                                        ;WITH INTERRUPT ENABLED
```

```
 9961
 9962  036176  104413                    WAT                     ;WAIT FOR DRY BIT TO SET
 9963  036200  002322                    RHDS1                   ;WAIT FOR RHDS1 REGISTER
 9964  036202  000200                    DRY                     ;WAIT FOR DRY BIT IN RHDS1 REGISTER
 9965  036204  004704                    2500.                   ;ALLOW 25000 MICRO SECONDS
 9966  036206  004704                    2500.                   ;DRY MUST SET BETWEEN
 9967                                                            ;00 AND 50000 MICRO SECONDS
 9968
 9969
 9970  036210  004737  041524            JSR     PC,@#CLDISK     ;SET R1-RHCS1, R2-RHCS2
 9971                                                            ;R3-RHDS1, R4-RHER1
 9972                                                            ;GIVE RH-11 INITIALIZE
 9973                                                            ;SETUP UNIT NUMBER
 9974
 9975                                    ;*10 WORDS OF EACH 12344,17777,0,52525,125252
 9976
 9977  036214  004037  041374            JSR     R0,@#CLAREA     ;CLEAR 10. WORDS, FROM WRFROM
 9978  036220  002470                    WRFROM                  ;STARTING FROM WRFROM
 9979  036222  000012                    10.                     ;10. WORDS
 9980  036224  012344                    <5*2000>!<7*40>!4                       ;FILL WITH <5*2000>!<7*40>!4
 9981
 9982  036226  004037  041374            JSR     R0,@#CLAREA     ;CLEAR 10. WORDS, FROM WRFROM+<10.*2>
 9983  036232  002514                    WRFROM+<10.*2>                  ;STARTING FROM WRFROM+<10.*2>
 9984  036234  000012                    10.                     ;10. WORDS
 9985  036236  177777                    -1                      ;FILL WITH -1
 9986
 9987  036240  004037  041374            JSR     R0,@#CLAREA     ;CLEAR 10. WORDS, FROM WRFROM+<20.*2>
 9988  036244  002540                    WRFROM+<20.*2>                  ;STARTING FROM WRFROM+<20.*2>
 9989  036246  000012                    10.                     ;10. WORDS
 9990  036250  000000                    0                       ;FILL WITH 0
 9991
 9992  036252  004037  041374            JSR     R0,@#CLAREA     ;CLEAR 10. WORDS, FROM WRFROM+<30.*2>
 9993  036256  002564                    WRFROM+<30.*2>                  ;STARTING FROM WRFROM+<30.*2>
 9994  036260  000012                    10.                     ;10. WORDS
 9995  036262  052525                    52525                   ;FILL WITH 52525
 9996
 9997  036264  004037  041374            JSR     R0,@#CLAREA     ;CLEAR 10. WORDS, FROM WRFROM+<40.*2>
 9998  036270  002610                    WRFROM+<40.*2>                  ;STARTING FROM WRFROM+<40.*2>
 9999  036272  000012                    10.                     ;10. WORDS
10000  036274  125252                    125252                  ;FILL WITH 125252
10001
10002
10003                                   ;*FILL LEFT ROTATING ZEROS FROM WRFROM+<50.*2>
10004
10005  036276  010146                    MOV     R1,-(SP)        ;;PUSH R1 ON STACK
10006  036300  012700  177776            MOV     #177776,R0      ;DATA
10007  036304  012705  000020            MOV     #16.,R5         ;COUNT
10008  036310  012701  002634            MOV     #WRFROM+<50.*2>,R1 ;WHERE DATA GOES
10009  036314  000261                    SEC
10010  036316  010021            1$:     MOV     R0,(R1)+        ;STORE DATA
10011  036320  006100                    ROL     R0              ;GET ZERO ONE BIT LEFT
10012  036322  005305                    DEC     R5              ;COUNT 16
10013  036324  001374                    BNE     1$              ;BRANCH IF 16 NOT DONE
10014
10015                                   ;*FILL LEFT ROTATING ONE INTO WRFROM+<65.*2>
10016
```

```
10017  036326  000241                          CLC
10018  036330  012700  000001                  MOV     #1,R0
10019  036334  010021                  2$:     MOV     R0,(R1)+
10020  036336  006300                          ASL     R0
10021  036340  103375                          BCC     2$
10022  036342  012601                          MOV     (SP)+,R1        ;;POP STACK INTO R1
10023
10024                                          ;*FILL REST OF DATA
10025
10026  036344  004037  041374                  JSR     R0,@#CLAREA     ;CLEAR 174. WORDS, FROM WRFROM+<82.*2>
10027  036350  002734                          WRFROM+<82.*2>                  ;STARTING FROM WRFROM+<82.*2>
10028  036352  000256                          174.                    ;174. WORDS
10029  036354  000377                          377                     ;FILL WITH 377
10030
10031  036356  004037  041374                  JSR     R0,@#CLAREA     ;CLEAR 2 WORDS, FROM WRFROM+<256.*2>
10032  036362  003470                          WRFROM+<256.*2>                 ;STARTING FROM WRFROM+<256.*2>
10033  036364  000002                          2                       ;2 WORDS
10034  036366  012345                          <5*2000>!<7*40>!5                       ;FILL WITH <5*2000>!<7*40>!5
10035
10036
10037                                          ;*FILL THE WRITE CHECK HEADER AND DATA
10038
10039  036370  004037  043476                  JSR     R0,@#RUN        ;SETUP TO RUN FOR DATA COMMAND
10040  036374  000005                          5                       ;CYLINDER 5
10041  036376     004                  .BYTE   4                       ;SECTOR 4
10042  036377     007                  .BYTE   7                       ;TRACK 7
10043  036400  177400                          -256.                   ;WORD COUNT = 256.
10044  036402  002470                          WRFROM                  ;BUS ADDRESS
10045                                                                  ;STARTING ADDRESS OF DATA
10046                                                                  ;BUFFER = WRFROM
10047  036404  000000                          0                       ;DO NOT INHIBIT BUS ADDRESS INCREMENT
10048  036406  014000                          ECI!FMT22                       ;16 BITS PER WORD FORMAT
10049                                                                  ;INHIBIT ECC CORRECTION
10050                                                                  ;DO NOT INHIBIT HEADER COMPARE
10051  036410  002436                          WRCHEK                  ;GET READY TO DO A WRCHEK
10052                                                  ;WRITE CHECK DATA WITH 50 IN RHCS1
10053
10054
10055                                          ;*SAVE REGISTERS FOR COMPARISON AFTER WRITE CHECK
10056  036412  004037  041672                  JSR     R0,@#SAVER      ;SAVE REGISTERS
10057  036416  002272                          RHWC                    ;RHWC IS THE FIRST REGISTER SAVED
10058  036420  004612                          SAVERE                  ;STARTING ADDRESS OF WHERE
10059                                                                  ;THE REGISTERS ARE SAVED
10060  036422  000022                          18.                     ;NUMBER OF REGISTERS
10061                                                                  ;SAVED = 18.
10062
10063  036424  004737  041604                  JSR     PC,@#CHECKT     ;CHECK DVA,RDY,MOL,DPR,DRY,VV = 1
10064                                                                  ;AND THAT NO STATUS BITS ARE STUCK = 1
10065  036430  104401  067033                  TYPE    ,CPHALT         ;CANNOT CONTINUE TESTING IF ANY OF
10066                                                                  ;THE FIRST SET OF BITS DON'T = 1
10067  036434  000000                          HALT                    ;STOP
10068
10069  036436  013777  004606  143622          MOV     @#RP4VEC,@RPVEC ;SET RP04 VECTOR ADDRESS
10070                                                                  ;TO 'TIME1' IF P-CLOCK IS PRESENT
10071                                                                  ;OR TO 'TIME2' IF P-CLOCK IS NOT PRESENT
10072                                                                  ;'TIME' WILL ONLY SAVE
```

```
10073                                                   ;CURRENT CYLINDER ADDRESS
10074                                                   ;AND LOOK AHEAD REGISTERS
10075
10076
10077  036444  013746  002436          MOV     @#WRCHEK,-(SP)  ;GET READY TO MOVE COMMAND
10078  036450  052716  000101          BIS     #GO!IE,(SP)     ;GET READY TO SET 'GO' AND
10079                                                   ;ENABLE INTERRUPT
10080  036454  012677  143620          MOV     (SP)+,@RHCS1    ;GO WITH
10081                                                    ;50 IN RHCS1 FOR WRITE CHECK DATA
10082                                                   ;WITH INTERRUPT ENABLED
10083  036460  011100                   MOV     @R1,R0          ;SAVE RHCS1 DURING ABOVE OPERATION
10084  036462  011305                   MOV     @R3,R5          ;SAVE RHDS1 DURING ABOVE OPERATION
10085
10086
10087  036464  104413                   WAT                     ;WAIT FOR RDY BIT TO SET
10088  036466  002300                   RHCS1                   ;WAIT FOR RHCS1 REGISTER
10089  036470  000200                   RDY                     ;WAIT FOR RDY BIT IN RHCS1 REGISTER
10090  036472  001732                   986.                    ;ALLOW 9860 MICRO SECONDS
10091  036474  001502                   834.                    ;RDY MUST SET BETWEEN
10092                                                   ;1520 AND 18200 MICRO SECONDS
10093
10094                                   ;*COMPARE CONTENTS OF RHCS1 AND RHDS1 ALREADY SAVED IN
10095                                   ;*R0 AND R5 IMMEDIATELY AFTER GO
10096
10097  036476  013746  002436          MOV     @#WRCHEK,-(SP)  ;SAVE COMMAND
10098  036502  052716  004101          BIS     #IE!DVA!GO,(SP) ;INCLUDE IE!DVA!GO
10099  036506  011637  001124          MOV     (SP),@#$GDDAT   ;SAVE FOR PRINTOUT
10100  036512  022600                   CMP     (SP)+,R0        ;DURING ABOVE OPERATION ONLY IE!DVA!GO
10101                                                   ;AND COMMAND SHOULD BE SET
10102  036514  001405                   BEQ     67$             ;BRANCH IF GOOD
10103  036516  010037  001126          MOV     R0,@#$BDDAT     ;BAD DATA
10104  036522  010137  004600          MOV     R1,@#REGADR     ;FAILING REGISTER RHCS1
10105  036526  104021                   ERROR   21              ;DURING ABOVE OPERATION ONLY
10106                                                   ;COMMAND AND IE!DVA!GO SHOULD BE SET
10107  036530  012746  010500  67$:     MOV     #MOL!DPR!VV,-(SP)       ;SAVE BITS SET DURING OPERATION IN RHDS1
10108  036534  011637  001124          MOV     (SP),@#$GDDAT   ;SAVE FOR PRINTOUT
10109  036540  022605                   CMP     (SP)+,R5        ;DURING ABOVE OPERATION ONLY MOL!DPR!VV
10110                                                   ;SHOULD BE SET
10111  036542  001405                   BEQ     69$             ;BRANCH IF GOOD
10112  036544  010537  001126          MOV     R5,@#$BDDAT     ;BAD DATA
10113  036550  010337  004600          MOV     R3,@#REGADR     ;FAILING REGISTER RHDS1
10114  036554  104063                   ERROR   63              ;DURING ABOVE OPERATION ONLY
10115                                                   ;MOL!DPR!VV SHOULD BE SET
10116  036556          69$:
10117
10118                                   ;*CHANGE SAVED REGISTERS TO EXPECTED VALUES
10119
10120  036556  004037  041426          JSR     R0,@#FILLRE     ;MOV 0 INTO SAVED RHWC
10121  036562  002272                   RHWC                    ;SAVED REGISTER TO CHANGE
10122  036564  000000                   0                       ;DATA
10123  036566  004037  041426          JSR     R0,@#FILLRE     ;MOV WRFROM+<256.*2> INTO SAVED RHBA
10124  036572  002274                   RHBA                    ;SAVED REGISTER TO CHANGE
10125  036574  003470                   WRFROM+<256.*2>                 ;DATA
10126  036576  004037  041426          JSR     R0,@#FILLRE     ;MOV 3405 INTO SAVED RHDST
10127  036602  002304                   RHDST                   ;SAVED REGISTER TO CHANGE
10128  036604  003405                   3405                    ;DATA
```

```
10129
10130                                   ;*COMPARE REGISTERS BEFORE WRITE CHECK HEADER AND DATA
10131                                   ;*WITH REGISTER AFTER COMMAND
10132
10133
10134  036606  004037  042512          JSR     R0,@#COMREG     ;COMPARE SAVED REGISTERS WITH
10135                                                          ;PRESENT VALUE
10136  036612  004612                  SAVERE                  ;GOOD DATA SAVED IN 'SAVERE'
10137  036614  002354                  WC                      ;TEST DATA STARTING FROM 'RHWC'
10138  036616  000022                  18.                     ;18. REGISTERS TO BE COMPARED
10139  036620  036624                  8$                      ;RETURN TO 8$ ON ERROR
10140  036622  036630                  9$                      ;RETURN TO 9$ ON NO ERROR
10141  036624  104040          8$:     ERROR   40              ;WRITE CHECK CAUSED
10142  036626  000207                  RTS     PC              ;AN IMPROPER REGISTER
10143                                                          ;CHANGE
10144                                                          ;GOOD DATA GIVES WHAT
10145                                                          ;SHOULD BE THERE
10146                                                          ;RECEIVED DATA GIVES WHAT
10147                                                          ;WAS THERE AFTER COMMANDS
10148
10149  036630                  9$:
10150
10151
```

```
10152
10153                                   ;;*****************************************************************
10154                                   ;*TEST 47       WRITE CHECK DATA USING UNIBUS B
10155                                   ;*      THIS TEST USES UNIBUS B IF CONNECTED TO THE RH
10156                                   ;*      IF UNIBUS B IS NOT CONNECTED THEN THIS TEST IS NOT PERFORMED
10157                                   ;*      THE DATA FOR THIS TEST IS WRITTEN ON DISK BY PREVIOUS TEST
10158                                   ;*      WRITE CHECK DATA CYLINDER 5, FORMAT 16 BITS PER WORDS
10159                                   ;*      TRACK 7, SECTOR 4, KEYS 0, NUMBER OF WORDS 258
10160                                   ;*      CONSISTING OF
10161                                   ;*      10 WORDS OF 12344 (6 BITS FOR CYL, 5 FOR TRACK, 5 FOR SECTOR)
10162                                   ;*      10 WORDS OF 177777
10163                                   ;*      10 WORDS OF 0
10164                                   ;*      10 WORDS OF 052525
10165                                   ;*      10 WORD OF 125252
10166                                   ;*      16 WORDS OF LEFT ROTATING ZERO (EG177776,177775)
10167                                   ;*      16 WORDS OF LEFT ROTATING ONE (EG 1,2,4,10)
10168                                   ;*      174 WORDS OF 377
10169                                   ;*      2 WORDS OF 12345
10170                                   ;*
10171                                   ;*
10172                                   ;*      FIRST THE ABOVE DATA IS FILLED INTO WRITE FROM BUFFER
10173                                   ;*      THEN THE ABOVE WRITE CHECK DATA IS GIVEN
10174                                   ;;*****************************************************************
10175  036630  000004                   TST47:  SCOPE
10176  036632  012706  001000                   MOV     #STACK,SP       ;RESET STACK
10177  036636  012737  000047  004604           MOV     #47,@#TSTNM     ;SAVE TEST NUMBER
10178
10179  036644  004737  041524                   JSR     PC,@#CLDISK     ;SET R1-RHCS1, R2-RHCS2
10180                                                                   ;R3-RHDS1, R4-RHER1
10181                                                                   ;GIVE RH-11 INITIALIZE
10182                                                                   ;SETUP UNIT NUMBER
10183
10184                                   ;*CHECK TO SEE IF THE PROGRAM IS RUNNING WITH AN RH70
10185
10186  036650  005737  004750                   TST     @#RH70          ;TEST FOR RH70 CONTROLLER
10187  036654  001402                           BEQ     30$             ;IF FLAG = 1, THIS TEST IS SKIPPED
10188  036656  000137  037376                   JMP     TST50   ;       JUMP TO NEXT TEST ----------)
10189  036662                           30$:                            ;IF FLAG = 1, DO THIS TEST
10190  036662  004737  041604                   JSR     PC,@#CHECKT     ;CHECK DVA,RDY,MOL,DPR,DRY,VV = 1
10191                                                                   ;AND THAT NO STATUS BITS ARE STUCK = 1
10192  036666  104401  067033                   TYPE    ,CPHALT         ;CANNOT CONTINUE TESTING IF ANY OF
10193                                                                   ;THE FIRST SET OF BITS DON'T = 1
10194  036672  000000                           HALT                    ;STOP
10195
10196  036674  005737  004732                   TST     @#UBUSB         ;IS UNIBUS B THERE
10197  036700  001402                           BEQ     11$             ;UNIBUS B THERE SO CONTINUE
10198  036702  000137  037376                   JMP     @#9$            ;NO UNIBUS B, GO TO NEXT TEST
10199
10200                                   ;*GET HEADS TO CYLINDER 5
10201
10202  036706                           11$:
10203
10204  036706  004037  041474                   JSR     R0,@#SEEKCY     ;SEEK FOR
10205  036712  000005                           5                       ;CYLINDER 5
10206
10207  036714  013777  004606  143344           MOV     @#RP4VEC,@RPVEC ;SET RP04 VECTOR ADDRESS
```

```
10208                                                   ;TO 'TIME1' IF P-CLOCK IS PRESENT
10209                                                   ;OR TO 'TIME2' IF P-CLOCK IS NOT PRESENT
10210                                                   ;'TIME' WILL ONLY SAVE
10211                                                   ;CURRENT CYLINDER ADDRESS
10212                                                   ;AND LOOK AHEAD REGISTERS
10213
10214
10215  036722  013746  002452          MOV     @#SEECOM,-(SP)  ;GET READY TO MOVE COMMAND
10216  036726  052716  000101          BIS     #GO!IE,(SP)     ;GET READY TO SET 'GO' AND
10217                                                   ;ENABLE INTERRUPT
10218  036732  012677  143342          MOV     (SP)+,@RHCS1    ;GO WITH
10219                                                    ;4 IN RHCS1 FOR SEEK
10220                                                   ;WITH INTERRUPT ENABLED
10221
10222
10223  036736  104413                  WAT                     ;WAIT FOR DRY BIT TO SET
10224  036740  002322                  RHDS1                   ;WAIT FOR RHDS1 REGISTER
10225  036742  000200                  DRY                     ;WAIT FOR DRY BIT IN RHDS1 REGISTER
10226  036744  004704                  2500.                   ;ALLOW 25000 MICRO SECONDS
10227  036746  004704                  2500.                   ;DRY MUST SET BETWEEN
10228                                                   ;00 AND 50000 MICRO SECONDS
10229
10230
10231  036750  004737  041524          JSR     PC,@#CLDISK     ;SET R1-RHCS1, R2-RHCS2
10232                                                   ;R3-RHDS1, R4-RHER1
10233                                                   ;GIVE RH-11 INITIALIZE
10234                                                   ;SETUP UNIT NUMBER
10235
10236                                  ;*10 WORDS OF EACH 12344,17777,0,52525,125252
10237
10238  036754  004037  041374          JSR     R0,@#CLAREA     ;CLEAR 10. WORDS, FROM WRFROM
10239  036760  002470                  WRFROM                  ;STARTING FROM WRFROM
10240  036762  000012                  10.                     ;10. WORDS
10241  036764  012344                  <5*2000>!<7*40>!4               ;FILL WITH <5*2000>!<7*40>!4
10242
10243  036766  004037  041374          JSR     R0,@#CLAREA     ;CLEAR 10. WORDS, FROM WRFROM+<10.*2>
10244  036772  002514                  WRFROM+<10.*2>                  ;STARTING FROM WRFROM+<10.*2>
10245  036774  000012                  10.                     ;10. WORDS
10246  036776  177777                  -1                      ;FILL WITH -1
10247
10248  037000  004037  041374          JSR     R0,@#CLAREA     ;CLEAR 10. WORDS, FROM WRFROM+<20.*2>
10249  037004  002540                  WRFROM+<20.*2>                  ;STARTING FROM WRFROM+<20.*2>
10250  037006  000012                  10.                     ;10. WORDS
10251  037010  000000                  0                       ;FILL WITH 0
10252
10253  037012  004037  041374          JSR     R0,@#CLAREA     ;CLEAR 10. WORDS, FROM WRFROM+<30.*2>
10254  037016  002564                  WRFROM+<30.*2>                  ;STARTING FROM WRFROM+<30.*2>
10255  037020  000012                  10.                     ;10. WORDS
10256  037022  052525                  52525                   ;FILL WITH 52525
10257
10258  037024  004037  041374          JSR     R0,@#CLAREA     ;CLEAR 10. WORDS, FROM WRFROM+<40.*2>
10259  037030  002610                  WRFROM+<40.*2>                  ;STARTING FROM WRFROM+<40.*2>
10260  037032  000012                  10.                     ;10. WORDS
10261  037034  125252                  125252                  ;FILL WITH 125252
10262
10263
```

```
10264                                          ;*FILL LEFT ROTATING ZEROS FROM WRFROM+<50.*2>
10265
10266  037036  010146                          MOV     R1,-(SP)        ;;PUSH R1 ON STACK
10267  037040  012700  177776                  MOV     #177776,R0      ;DATA
10268  037044  012705  000020                  MOV     #16.,R5         ;COUNT
10269  037050  012701  002634                  MOV     #WRFROM+<50.*2>,R1 ;WHERE DATA GOES
10270  037054  000261                          SEC
10271  037056  010021                  1$:     MOV     R0,(R1)+        ;STORE DATA
10272  037060  006100                          ROL     R0              ;GET ZERO ONE BIT LEFT
10273  037062  005305                          DEC     R5              ;COUNT 16
10274  037064  001374                          BNE     1$              ;BRANCH IF 16 NOT DONE
10275
10276                                          ;*FILL LEFT ROTATING ONE INTO WRFROM+<65.*2>
10277
10278  037066  000241                          CLC
10279  037070  012700  000001                  MOV     #1,R0
10280  037074  010021                  2$:     MOV     R0,(R1)+
10281  037076  006300                          ASL     R0
10282  037100  103375                          BCC     2$
10283  037102  012601                          MOV     (SP)+,R1        ;;POP STACK INTO R1
10284
10285                                          ;*FILL REST OF DATA
10286  037104  004037  041374                  JSR     R0,@#CLAREA     ;CLEAR 174. WORDS, FROM WRFROM+<82.*2>
10287  037110  002734                          WRFROM+<82.*2>                  ;STARTING FROM WRFROM+<82.*2>
10288  037112  000256                          174.                    ;174. WORDS
10289  037114  000377                          377                     ;FILL WITH 377
10290
10291  037116  004037  041374                  JSR     R0,@#CLAREA     ;CLEAR 2 WORDS, FROM WRFROM+<256.*2>
10292  037122  003470                          WRFROM+<256.*2>                 ;STARTING FROM WRFROM+<256.*2>
10293  037124  000002                          2                       ;2 WORDS
10294  037126  012345                          <5*2000>!<7*40>!5                       ;FILL WITH <5*2000>!<7*40>!5
10295
10296
10297                                          ;*FILL THE WRITE CHECK HEADER AND DATA
10298
10299  037130  004037  043476                  JSR     R0,@#RUN        ;SETUP TO RUN FOR DATA COMMAND
10300  037134  000005                          5                       ;CYLINDER 5
10301  037136     004                  .BYTE   4                       ;SECTOR 4
10302  037137     007                  .BYTE   7                       ;TRACK 7
10303  037140  177400                          -256.                   ;WORD COUNT = 256.
10304  037142  002470                          WRFROM                  ;BUS ADDRESS
10305                                                                  ;STARTING ADDRESS OF DATA
10306                                                                  ;BUFFER = WRFROM
10307  037144  000000                          0                       ;DO NOT INHIBIT BUS ADDRESS INCREMENT
10308  037146  014000                          ECI!FMT22                       ;16 BITS PER WORD FORMAT
10309                                                                  ;INHIBIT ECC CORRECTION
10310                                                                  ;DO NOT INHIBIT HEADER COMPARE
10311  037150  002436                          WRCHEK                  ;GET READY TO DO A WRCHEK
10312                                                  ;WRITE CHECK DATA WITH 50 IN RHCS1
10313
10314
10315
10316  037152  052777  002000  143120          BIS     #PSEL,@RHCS1    ;SET PORT B
10317                                                                  ;THAT IS UNIBUS B
10318
10319                                          ;*SAVE REGISTERS FOR COMPARISON AFTER WRITE CHECK
```

```
10320  037160  004037  041672              JSR     R0,@#SAVER      ;SAVE REGISTERS
10321  037164  002272                      RHWC                    ;RHWC IS THE FIRST REGISTER SAVED
10322  037166  004612                      SAVERE                  ;STARTING ADDRESS OF WHERE
10323                                                               ;THE REGISTERS ARE SAVED
10324  037170  000022                      18.                     ;NUMBER OF REGISTERS
10325                                                               ;SAVED = 18.
10326
10327  037172  004737  041604              JSR     PC,@#CHECKT     ;CHECK DVA,RDY,MOL,DPR,DRY,VV = 1
10328                                                               ;AND THAT NO STATUS BITS ARE STUCK = 1
10329  037176  104401  067033              TYPE    ,CPHALT         ;CANNOT CONTINUE TESTING IF ANY OF
10330                                                               ;THE FIRST SET OF BITS DON'T = 1
10331  037202  000000                      HALT                    ;STOP
10332
10333  037204  013777  004606  143054      MOV     @#RP4VEC,@RPVEC ;SET RP04 VECTOR ADDRESS
10334                                                               ;TO 'TIME1' IF P-CLOCK IS PRESENT
10335                                                               ;OR TO 'TIME2' IF P-CLOCK IS NOT PRESENT
10336                                                               ;'TIME' WILL ONLY SAVE
10337                                                               ;CURRENT CYLINDER ADDRESS
10338                                                               ;AND LOOK AHEAD REGISTERS
10339
10340                                      ;*SET PORT SELECT
10341
10342  037212  013746  002436              MOV     @#WRCHEK,-(SP)  ;GET READY TO MOVE COMMAND
10343  037216  052716  002101              BIS     #GO!IE!PSEL,(SP)        ;GET READY TO SET 'GO' AND
10344                                                               ;ENABLE INTERRUPT
10345  037222  012677  143052              MOV     (SP)+,@RHCS1    ;GO WITH
10346                                                               ;50 IN RHCS1 FOR WRITE CHECK DATA
10347                                                               ;WITH INTERRUPT ENABLED
10348  037226  011100                      MOV     @R1,R0          ;SAVE RHCS1 DURING ABOVE OPERATION
10349  037230  011305                      MOV     @R3,R5          ;SAVE RHDS1 DURING ABOVE OPERATION
10350
10351
10352  037232  104413                      WAT                     ;WAIT FOR RDY BIT TO SET
10353  037234  002300                      RHCS1                   ;WAIT FOR RHCS1 REGISTER
10354  037236  000200                      RDY                     ;WAIT FOR RDY BIT IN RHCS1 REGISTER
10355  037240  001732                      986.                    ;ALLOW 9860 MICRO SECONDS
10356  037242  001502                      834.                    ;RDY MUST SET BETWEEN
10357                                                               ;1520 AND 18200 MICRO SECONDS
10358
10359                                      ;*COMPARE CONTENTS OF RHCS1 AND RHDS1 ALREADY SAVED IN
10360                                      ;*R0 AND R5 IMMEDIATELY AFTER GO
10361
10362  037244  013746  002436              MOV     @#WRCHEK,-(SP)  ;SAVE COMMAND
10363  037250  052716  006101              BIS     #IE!DVA!PSEL!GO,(SP)    ;INCLUDE IE!DVA!PSEL!GO
10364  037254  011637  001124              MOV     (SP),@#$GDDAT   ;SAVE FOR PRINTOUT
10365  037260  022600                      CMP     (SP)+,R0        ;DURING ABOVE OPERATION ONLY IE!DVA!PSEL!GO
10366                                                               ;AND COMMAND SHOULD BE SET
10367  037262  001405                      BEQ     67$             ;BRANCH IF GOOD
10368  037264  010037  001126              MOV     R0,@#$BDDAT     ;BAD DATA
10369  037270  010137  004600              MOV     R1,@#REGADR     ;FAILING REGISTER RHCS1
10370  037274  104021                      ERROR   21              ;DURING ABOVE OPERATION ONLY
10371                                                               ;COMMAND AND IE!DVA!PSEL!GO SHOULD BE SET
10372  037276  012746  010500      67$:    MOV     #MOL!DPR!VV,-(SP)       ;SAVE BITS SET DURING OPERATION IN RHDS1
10373  037302  011637  001124              MOV     (SP),@#$GDDAT   ;SAVE FOR PRINTOUT
10374  037306  022605                      CMP     (SP)+,R5        ;DURING ABOVE OPERATION ONLY MOL!DPR!VV
10375                                                               ;SHOULD BE SET
```

```
10376  037310  001405                    BEQ     69$                ;BRANCH IF GOOD
10377  037312  010537  001126            MOV     R5,@#$BDDAT        ;BAD DATA
10378  037316  010337  004600            MOV     R3,@#REGADR        ;FAILING REGISTER RHDS1
10379  037322  104063                    ERROR   63                 ;DURING ABOVE OPERATION ONLY
10380                                                               ;MOL!DPR!VV SHOULD BE SET
10381  037324                    69$:
10382
10383                                    ;*CHANGE SAVED REGISTERS TO EXPECTED VALUES
10384
10385  037324  004037  041426            JSR     R0,@#FILLRE        ;MOV 0 INTO SAVED RHWC
10386  037330  002272                    RHWC                       ;SAVED REGISTER TO CHANGE
10387  037332  000000                    0                          ;DATA
10388  037334  004037  041426            JSR     R0,@#FILLRE        ;MOV WRFROM+<256.*2> INTO SAVED RHBA
10389  037340  002274                    RHBA                       ;SAVED REGISTER TO CHANGE
10390  037342  003470                    WRFROM+<256.*2>                    ;DATA
10391  037344  004037  041426            JSR     R0,@#FILLRE        ;MOV 3405 INTO SAVED RHDST
10392  037350  002304                    RHDST                      ;SAVED REGISTER TO CHANGE
10393  037352  003405                    3405                       ;DATA
10394
10395                                    ;*COMPARE REGISTERS BEFORE WRITE CHECK HEADER AND DATA
10396                                    ;*WITH REGISTER AFTER COMMAND
10397
10398  037354  004037  042512            JSR     R0,@#COMREG        ;COMPARE SAVED REGISTERS WITH
10399                                                               ;PRESENT VALUE
10400  037360  004612                    SAVERE                     ;GOOD DATA SAVED IN 'SAVERE'
10401  037362  002354                    WC                         ;TEST DATA STARTING FROM 'RHWC'
10402  037364  000022                    18.                        ;18. REGISTERS TO BE COMPARED
10403  037366  037372                    8$                         ;RETURN TO 8$ ON ERROR
10404  037370  037376                    9$                         ;RETURN TO 9$ ON NO ERROR
10405
10406  037372  104076            8$:     ERROR   76                 ;WHILE USING UNIBUS B
10407                                                               ;WRITE CHECK CAUSED
10408  037374  000207                    RTS     PC                 ;AN IMPROPER REGISTER
10409                                                               ;CHANGE
10410                                                               ;GOOD DATA GIVES WHAT
10411                                                               ;SHOULD BE THERE
10412                                                               ;RECEIVED DATA GIVES WHAT
10413                                                               ;WAS THERE AFTER COMMANDS
10414
10415  037376                    9$:
10416
10417
10418
10419
10420
```

```
10421
10422                                   ;;*****************************************************************
10423                                   ;*TEST 50       WRITE PROTECT OPERATION
10424                                   ;*      IF STARTING ADDRESS 220 IS USED THIS TEST WILL NOT BE PERFORMED
10425                                   ;*
10426                                   ;*      IF THE PROGRAM WORKS UNDER ACT-11 MONITOR
10427                                   ;*      THEN THIS TEST IS NOT PERFORMED
10428                                   ;*
10429                                   ;*      IF NO ACT-11 MONITOR IS PRESENT
10430                                   ;*      THEN THIS TEST IS PERFORMED ONLY ON THE FIRST PASS
10431                                   ;*      ON SUBSEQUENT PASSES THIS TEST IS NOT DONE
10432                                   ;*
10433                                   ;*      WRITE FROM BUFFER IS FILLED WITH ALL ONES AND
10434                                   ;*      SECTOR 0, TRACK 0, CYLINDER 0 IS FILLED WITH
10435                                   ;*      ALL ONES
10436                                   ;*      ALL REGISTERS ARE SAVED THEN WRITE LOCK BUTTON IS
10437                                   ;*      PRESSED AND ALL REGISTERS ARE CHECKED.
10438                                   ;*      WRITE FROM BUFFER IS FILLED WITH 377 AND A WRITE IS
10439                                   ;*      ATTEMPTED TO SECTOR 0, TRACK 0, CYLINDER 0  70. WORDS
10440                                   ;*      ALL REGISTERS ARE CHECKED
10441                                   ;*      THE SAME SECTOR IS READ AND DATA COMPARED TO SEE
10442                                   ;*      THAT NOTHING GOT DESTROYED (READ DATA SHOULD BE ALL
10443                                   ;*      ONES AND NOT 377)
10444                                   ;*      THEN WRITE LOCK BUTTON IS PRESSED TO UNLOCK
10445                                   ;*      WRITE LOCKS AND ALL REGISTERS ARE COMPARED
10446
10447                                   ;;*****************************************************************
10448  037376  000004                 TST50:  SCOPE
10449
10450
10451                                           ;*THIS CODE CHECKS TO SEE IF MANUAL INTERVENTION TESTS ARE OK
10452
10453  037400  005737  004724                   TST     @#NOPUSH        ;IS THIS A 220 START ?
10454  037404  001007                           BNE     1$              ;SKIP THIS TEST IF SO
10455  037406  005737  000042                   TST     @#42            ;MONITOR (ACT 11) RETURN ADDRESS ?
10456  037412  001004                           BNE     1$              ;SKIP THIS TEST
10457  037414  005737  001100                   TST     @#$PASS         ;FIRST PASS ?
10458  037420  001001                           BNE     1$              ;SKIP THIS TEST IF NOT
10459  037422  000402                           BR      2$              ;CONTINUE WITH THIS TEST
10460
10461  037424                          1$:
10462  037424  000137  040622                   JMP     TST51   ;       JUMP TO NEXT TEST ----------------)
10463  037430                          2$:
10464  037430  012706  001000                   MOV     #STACK,SP       ;RESET STACK
10465  037434  012737  000050  004604           MOV     #50,@#TSTNM     ;SAVE TEST NUMBER
10466
10467  037442  004737  041524                   JSR     PC,@#CLDISK     ;SET R1-RHCS1, R2-RHCS2
10468                                                                   ;R3-RHDS1, R4-RHER1
10469                                                                   ;GIVE RH-11 INITIALIZE
10470                                                                   ;SETUP UNIT NUMBER
10471
10472                                           ;*FILL SECTOR 0, TRACK 0, CYL 0 WITH ONES
10473                                           ;*FILL WRITE FROM BUFFER
10474  037446  004037  041374                   JSR     R0,@#CLAREA     ;CLEAR 256. WORDS, FROM WRFROM
10475  037452  002470                           WRFROM                  ;STARTING FROM WRFROM
10476  037454  000400                           256.                    ;256. WORDS
```

```
10477  037456  177777                          -1                          ;FILL WITH -1
10478
10479
10480                                          ;*FILL WRITE DATA COMMAND
10481
10482  037460  004037  043476                  JSR     R0,@#RUN            ;SETUP TO RUN FOR DATA COMMAND
10483  037464  000000                          0                           ;CYLINDER 0
10484  037466     000                  .BYTE   0                           ;SECTOR 0
10485  037467     000                  .BYTE   0                           ;TRACK 0
10486  037470  177400                          -256.                       ;WORD COUNT = 256.
10487  037472  002470                          WRFROM                      ;BUS ADDRESS
10488                                                                      ;STARTING ADDRESS OF DATA
10489                                                                      ;BUFFER = WRFROM
10490  037474  000000                          0                           ;DO NOT INHIBIT BUS ADDRESS INCREMENT
10491  037476  010000                          FMT22                       ;16 BITS PER WORD FORMAT
10492                                                                      ;DO NOT INHIBIT ECC CORRECTION
10493                                                                      ;DO NOT INHIBIT HEADER COMPARE
10494  037500  002442                          WRIDAT                      ;GET READY TO DO A WRIDAT
10495                                                  ;WRITE DATA WITH 60 IN RHCS1
10496
10497
10498  037502  004737  041604                  JSR     PC,@#CHECKT         ;CHECK DVA,RDY,MOL,DPR,DRY,VV = 1
10499                                                                      ;AND THAT NO STATUS BITS ARE STUCK = 1
10500  037506  104401  067033                  TYPE    ,CPHALT             ;CANNOT CONTINUE TESTING IF ANY OF
10501                                                                      ;THE FIRST SET OF BITS DON'T = 1
10502  037512  000000                          HALT                        ;STOP
10503
10504  037514  013777  004606  142544          MOV     @#RP4VEC,@RPVEC     ;SET RP04 VECTOR ADDRESS
10505                                                                      ;TO 'TIME1' IF P-CLOCK IS PRESENT
10506                                                                      ;OR TO 'TIME2' IF P-CLOCK IS NOT PRESENT
10507                                                                      ;'TIME' WILL ONLY SAVE
10508                                                                      ;CURRENT CYLINDER ADDRESS
10509                                                                      ;AND LOOK AHEAD REGISTERS
10510
10511
10512  037522  013746  002442                  MOV     @#WRIDAT,-(SP)      ;GET READY TO MOVE COMMAND
10513  037526  052716  000101                  BIS     #GO!IE,(SP)         ;GET READY TO SET 'GO' AND
10514                                                                      ;ENABLE INTERRUPT
10515  037532  012677  142542                  MOV     (SP)+,@RHCS1        ;GO WITH
10516                                                                       ;60 IN RHCS1 FOR WRITE DATA
10517                                                                      ;WITH INTERRUPT ENABLED
10518
10519                                          ;*TIME IS NOT CRITICAL
10520
10521  037536  104413                          WAT                         ;WAIT FOR RDY BIT TO SET
10522  037540  002300                          RHCS1                       ;WAIT FOR RHCS1 REGISTER
10523  037542  000200                          RDY                         ;WAIT FOR RDY BIT IN RHCS1 REGISTER
10524  037544  004704                          2500.                       ;ALLOW 25000 MICRO SECONDS
10525  037546  004704                          2500.                       ;RDY MUST SET BETWEEN
10526                                                                      ;00 AND 50000 MICRO SECONDS
10527
10528                                          ;*SAVE REGISTERS FOR COMPARISON AFTER WRITE PROTECT
10529                                          ;*BUTTON HAS BEEN HIT
10530
10531  037550  004037  041672                  JSR     R0,@#SAVER          ;SAVE REGISTERS
10532  037554  002272                          RHWC                        ;RHWC IS THE FIRST REGISTER SAVED
```

```
10533  037556  004612                          SAVERE                  ;STARTING ADDRESS OF WHERE
10534                                                                   ;THE REGISTERS ARE SAVED
10535  037560  000022                          18.                     ;NUMBER OF REGISTERS
10536                                                                   ;SAVED = 18.
10537
10538  037562  104401  037570                  TYPE    ,68$            ;;TYPE ASCIZ STRING
10539  037566  000407                          BR      67$             ;;GET OVER THE ASCIZ
10540                                  ;;68$:  .ASCIZ  <15><12>/ON DRIVE  /
10541  037606                          67$:
10542  037606  013746  004716                  MOV     @#UNIT,-(SP)            ;GET UNIT UNDER TEST
10543  037612  104405                          TYPDS
10544  037614  104401  037622                  TYPE    ,70$            ;;TYPE ASCIZ STRING
10545  037620  000442                          BR      69$             ;;GET OVER THE ASCIZ
10546                                  ;;70$:  .ASCIZ  <15><12>/PUSH WRITE PROTECT BUTTON TO LOCK OUT WRITES THEN HIT CONTINUE/
10547  037726                          69$:
10548  037726  000000                          HALT
10549                                          ;*THE ONLY REGISTER THAT SHOULD  CHANGE IS RHDS1 - BIT #11
10550                                          ;*-WRL
10551
10552  037730  004037  042404                  JSR     R0,@#CHREG      ;CHANGE BITS IN SAVED REGISTER
10553  037734  002322                          RHDS1                   ;CHANGE RHDS1 REGISTER
10554
10555  037736  000001                          1                       ;1 BIT/BITS TO BE CHANGED
10556  037740  000001                          1                       ;NEW VALUE OF WRL IS 1
10557  037742  004000                          WRL                     ;CHANGE WRL BIT
10558
10559                                          ;*COMPARE ALL REGISTERS BEFORE WRITE WAS LOCKED
10560                                          ;*OUT WITH REGISTER VALUES AFTER WRITE WAS LOCKED OUT
10561
10562  037744  004037  042512                  JSR     R0,@#COMREG     ;COMPARE SAVED REGISTERS WITH
10563                                                                  ;PRESENT VALUE
10564  037750  004612                          SAVERE                  ;GOOD DATA SAVED IN 'SAVERE'
10565  037752  002354                          WC                      ;TEST DATA STARTING FROM 'RHWC'
10566  037754  000022                          18.                     ;18. REGISTERS TO BE COMPARED
10567  037756  037762                          3$                      ;RETURN TO 3$ ON ERROR
10568  037760  037766                          4$                      ;RETURN TO 4$ ON NO ERROR
10569
10570  037762  104041                  3$:     ERROR   41              ;LOCKING OUT WRITE BY
10571  037764  000207                          RTS     PC              ;WRITE LOCK BUTTON CAUSED
10572                                                                  ;IMPROPER REGISTER CHANGE
10573                                                                  ;GOOD DATA GIVES WHAT SHOULD
10574                                                                  ;BE THERE
10575                                                                  ;RECEIVED DATA GIVES WHAT
10576                                                                  ;WAS THERE AFTER WRITE
10577                                                                  ;WAS LOCKED OUT BY BUTTON
10578
10579                                          ;*NOW A WRITE WILL BE ATTEMPTED WITH WRITE LOCKED
10580                                          ;*OUT BY BUTTON
10581                                          ;*FILL WRITE FROM BUFFER WITH 377
10582
10583  037766  013737  037766  001110  4$:     MOV     @#4$,@#$LPERR   ;SCOPE LOOP STARTS FROM HERE
10584
10585
10586  037774  004737  041524                  JSR     PC,@#CLDISK     ;SET R1-RHCS1, R2-RHCS2
10587                                                                  ;R3-RHDS1, R4-RHER1
10588                                                                  ;GIVE RH-11 INITIALIZE
```

```
10589
10590  040000  004037  041374                JSR     R0,@#CLAREA     ;SETUP UNIT NUMBER
10591  040004  002470                       WRFROM                  ;CLEAR 256. WORDS, FROM WRFROM
10592  040006  000400                       256.                    ;STARTING FROM WRFROM
10593  040010  000377                       377                     ;256. WORDS
10594                                                                ;FILL WITH 377
10595
10596                                        ;*TRY A ONE WORD WRITE
10597
10598  040012  004037  043476                JSR     R0,@#RUN        ;SETUP TO RUN FOR DATA COMMAND
10599  040016  000000                       0                       ;CYLINDER 0
10600  040020     000               .BYTE   0                       ;SECTOR 0
10601  040021     000               .BYTE   0                       ;TRACK 0
10602  040022  177777                       -1.                     ;WORD COUNT = 1.
10603  040024  002470                       WRFROM                  ;BUS ADDRESS
10604                                                                ;STARTING ADDRESS OF DATA
10605                                                                ;BUFFER = WRFROM
10606  040026  000000                       0                       ;DO NOT INHIBIT BUS ADDRESS INCREMENT
10607  040030  010000                       FMT22                   ;16 BITS PER WORD FORMAT
10608                                                                ;DO NOT INHIBIT ECC CORRECTION
10609                                                                ;DO NOT INHIBIT HEADER COMPARE
10610  040032  002442                       WRIDAT                  ;GET READY TO DO A WRIDAT
10611                                                    ;WRITE DATA WITH 60 IN RHCS1
10612
10613
10614                                        ;*SAVE REGISTERS
10615  040034  004037  041672                JSR     R0,@#SAVER      ;SAVE REGISTERS
10616  040040  002272                       RHWC                    ;RHWC IS THE FIRST REGISTER SAVED
10617  040042  004612                       SAVERE                  ;STARTING ADDRESS OF WHERE
10618                                                                ;THE REGISTERS ARE SAVED
10619  040044  000022                       18.                     ;NUMBER OF REGISTERS
10620                                                                ;SAVED = 18.
10621
10622  040046  013777  004606  142212        MOV     @#RP4VEC,@RPVEC ;SET RP04 VECTOR ADDRESS
10623                                                                ;TO 'TIME1' IF P-CLOCK IS PRESENT
10624                                                                ;OR TO 'TIME2' IF P-CLOCK IS NOT PRESENT
10625                                                                ;'TIME' WILL ONLY SAVE
10626                                                                ;CURRENT CYLINDER ADDRESS
10627                                                                ;AND LOOK AHEAD REGISTERS
10628
10629
10630  040054  013746  002442                MOV     @#WRIDAT,-(SP)  ;GET READY TO MOVE COMMAND
10631  040060  052716  000101                BIS     #GO!IE,(SP)     ;GET READY TO SET 'GO' AND
10632                                                                ;ENABLE INTERRUPT
10633  040064  012677  142210                MOV     (SP)+,@RHCS1    ;GO WITH
10634                                                                 ;60 IN RHCS1 FOR WRITE DATA
10635                                                                ;WITH INTERRUPT ENABLED
10636                                        ;*TIME IS NOT CRITICAL
10637
10638  040070  104413                       WAT                     ;WAIT FOR RDY BIT TO SET
10639  040072  002300                       RHCS1                   ;WAIT FOR RHCS1 REGISTER
10640  040074  000200                       RDY                     ;WAIT FOR RDY BIT IN RHCS1 REGISTER
10641  040076  004704                       2500.                   ;ALLOW 25000 MICRO SECONDS
10642  040100  004704                       2500.                   ;RDY MUST SET BETWEEN
10643                                                                ;00 AND 50000 MICRO SECONDS
10644
```

```
10645                                           ;*CHANGE SAVED REGISTERS TO EXPECTED VALUE
10646
10647  040102  017737  142164  004612           MOV     @RHWC,@#SAVERE  ;RHWC IS UNPREDICTABLE
10648  040110  017737  142160  004614           MOV     @RHBA,@#SAVERE+2;RHBA IS UNPREDICTABLE
10649  040116  017746  142154                   MOV     @RHCS2,-(SP)    ;GET RHCS2
10650  040122  042716  177477                   BIC     #^C<IR!OR>,(SP) ;KEEP IR AND OR
10651  040126  042737  000300  004616           BIC     #IR!OR,@#SAVERE+4;CLEAR SAVED IR OR
10652  040134  052637  004616                   BIS     (SP)+,@#SAVERE+4;SET OR IR AS REQUIRED
10653
10654  040140  004037  042404                   JSR     R0,@#CHREG      ;CHANGE BITS IN SAVED REGISTER
10655  040144  002300                           RHCS1                   ;CHANGE RHCS1 REGISTER
10656
10657  040146  000002                           2                       ;2 BIT/BITS TO BE CHANGED
10658  040150  000001                           1                       ;NEW VALUE OF SC IS 1
10659  040152  100000                           SC                      ;CHANGE SC BIT
10660  040154  000001                           1                       ;NEW VALUE OF TRE IS 1
10661  040156  040000                           TRE                     ;CHANGE TRE BIT
10662  040160  004037  041426                   JSR     R0,@#FILLRE     ;MOV 1 INTO SAVED RHDST
10663  040164  002304                           RHDST                   ;SAVED REGISTER TO CHANGE
10664  040166  000001                           1                       ;DATA
10665
10666  040170  004037  042404                   JSR     R0,@#CHREG      ;CHANGE BITS IN SAVED REGISTER
10667  040174  002302                           RHER1                   ;CHANGE RHER1 REGISTER
10668
10669  040176  000001                           1                       ;1 BIT/BITS TO BE CHANGED
10670  040200  000001                           1                       ;NEW VALUE OF WLE IS 1
10671  040202  004000                           WLE                     ;CHANGE WLE BIT
10672
10673  040204  004037  042404                   JSR     R0,@#CHREG      ;CHANGE BITS IN SAVED REGISTER
10674  040210  002322                           RHDS1                   ;CHANGE RHDS1 REGISTER
10675
10676  040212  000002                           2                       ;2 BIT/BITS TO BE CHANGED
10677  040214  000001                           1                       ;NEW VALUE OF ATA IS 1
10678  040216  100000                           ATA                     ;CHANGE ATA BIT
10679  040220  000001                           1                       ;NEW VALUE OF ERR IS 1
10680  040222  040000                           ERR                     ;CHANGE ERR BIT
10681
10682  040224  053737  004740  004636           BIS     @#ATTENT,@#SAVERE+24    ;SET APPROPRIATE 'ATA' BITS
10683                                                                   ;FOR WORKING DRIVE IN
10684                                                                   ;SAVED RHAS LOACTION
10685
10686                                           ;*COMPARE REGISTERS BEFORE WRITE WAS ATTEMPTED
10687                                           ;*WITH REGISTERS AFTER ATTEMPT
10688
10689
10690  040232  004037  042512                   JSR     R0,@#COMREG     ;COMPARE SAVED REGISTERS WITH
10691                                                                   ;PRESENT VALUE
10692  040236  004612                           SAVERE                  ;GOOD DATA SAVED IN 'SAVERE'
10693  040240  002354                           WC                      ;TEST DATA STARTING FROM 'RHWC'
10694  040242  000022                           18.                     ;18. REGISTERS TO BE COMPARED
10695  040244  040250                           5$                      ;RETURN TO 5$ ON ERROR
10696  040246  040254                           6$                      ;RETURN TO 6$ ON NO ERROR
10697
10698  040250  104042                   5$:     ERROR   42              ;ATTEMPTING TO WRITE WITH
10699  040252  000207                           RTS     PC              ;WRITE LOCKED OUT
10700                                                                   ;CAUSED IMPROPER REGISTER
```

```
10701                                                   ;CHANGE
10702                                                   ;GOOD DATA GIVES WHAT SHOULD
10703                                                   ;BE THERE
10704                                                   ;RECEIVED DATA GIVES WHAT WAS
10705                                                   ;THERE AFTER ATTEMPT
10706
10707                                   ;*NOW A READ WILL BE DONE TO DETERMIN THAT
10708                                   ;*READS CAN BE DONE WITH WRITE LOCKED OUT AND
10709                                   ;*THAT NO DATA ON DISK GOT CHANGED, BUT FIRST CLEAR ERROR
10710
10711  040254                       6$:
10712
10713  040254  004737  041524              JSR     PC,@#CLDISK     ;SET R1-RHCS1, R2-RHCS2
10714                                                               ;R3-RHDS1, R4-RHER1
10715                                                               ;GIVE RH-11 INITIALIZE
10716                                                               ;SETUP UNIT NUMBER
10717
10718                                   ;*FILL READ INTO BUFFER WITH 0
10719  040260  004037  041374              JSR     R0,@#CLAREA     ;CLEAR 256. WORDS, FROM REINTO
10720  040264  003534                      REINTO                  ;STARTING FROM REINTO
10721  040266  000400                      256.                    ;256. WORDS
10722  040270  000000                      0                       ;FILL WITH 0
10723
10724
10725                                   ;*FILL WRITE FROM BUFFER WITH EXPECTED DATA FROM READ
10726  040272  004037  041374              JSR     R0,@#CLAREA     ;CLEAR 256. WORDS, FROM WRFROM
10727  040276  002470                      WRFROM                  ;STARTING FROM WRFROM
10728  040300  000400                      256.                    ;256. WORDS
10729  040302  177777                      -1                      ;FILL WITH -1
10730
10731
10732                                   ;*FILL COMMAND
10733
10734  040304  004037  043476              JSR     R0,@#RUN        ;SETUP TO RUN FOR DATA COMMAND
10735  040310  000000                      0                       ;CYLINDER 0
10736  040312     000          .BYTE       0                       ;SECTOR 0
10737  040313     000          .BYTE       0                       ;TRACK 0
10738  040314  177400                      -256.                   ;WORD COUNT = 256.
10739  040316  003534                      REINTO                  ;BUS ADDRESS
10740                                                               ;STARTING ADDRESS OF DATA
10741                                                               ;BUFFER = REINTO
10742  040320  000000                      0                       ;DO NOT INHIBIT BUS ADDRESS INCREMENT
10743  040322  014000                      ECI!FMT22                       ;16 BITS PER WORD FORMAT
10744                                                               ;INHIBIT ECC CORRECTION
10745                                                               ;DO NOT INHIBIT HEADER COMPARE
10746  040324  002446                      READAT                  ;GET READY TO DO A READAT
10747                                                       ;READ DATA WITH 70 IN RHCS1
10748
10749
10750  040326  013777  004606  141732      MOV     @#RP4VEC,@RPVEC ;SET RP04 VECTOR ADDRESS
10751                                                               ;TO 'TIME1' IF P-CLOCK IS PRESENT
10752                                                               ;OR TO 'TIME2' IF P-CLOCK IS NOT PRESENT
10753                                                               ;'TIME' WILL ONLY SAVE
10754                                                               ;CURRENT CYLINDER ADDRESS
10755                                                               ;AND LOOK AHEAD REGISTERS
10756
```

```
10757
10758  040334  013746  002446                  MOV     @#READAT,-(SP)  ;GET READY TO MOVE COMMAND
10759  040340  052716  000101                  BIS     #GO!IE,(SP)     ;GET READY TO SET 'GO' AND
10760                                                                    ;ENABLE INTERRUPT
10761  040344  012677  141730                  MOV     (SP)+,@RHCS1    ;GO WITH
10762                                                                     ;70 IN RHCS1 FOR READ DATA
10763                                                                    ;WITH INTERRUPT ENABLED
10764
10765
10766  040350  104413                          WAT                     ;WAIT FOR RDY BIT TO SET
10767  040352  002300                          RHCS1                   ;WAIT FOR RHCS1 REGISTER
10768  040354  000200                          RDY                     ;WAIT FOR RDY BIT IN RHCS1 REGISTER
10769  040356  001614                          908.                    ;ALLOW 9080 MICRO SECONDS
10770  040360  001507                          839.                    ;RDY MUST SET BETWEEN
10771                                                                    ;690 AND 17470 MICRO SECONDS
10772
10773                                          ;*COMPARE READ DATA
10774
10775  040362  004037  043542                  JSR     R0,@#COMPAR     ;COMPARE TWO BLOCKS OF MEMORY
10776  040366  002470                          WRFROM                  ;GOOD DATA STARTS FROM WRFROM
10777  040370  003534                          REINTO                  ;TEST DATA STARTS FROM REINTO
10778  040372  000400                          256.                    ;256. WORDS TO BE COMPARED
10779  040374  040400                          7$                      ;RETURN TO 7$ ON ERROR
10780  040376  040402                          8$                      ;RETURN TO 8$ ON NO ERROR
10781
10782
10783  040400  104043                  7$:     ERROR   43              ;WRITING WITH WRITE
10784                                                                    ;LOCKED CHANGED DISK
10785                                                                    ;GOOD DATA GIVES WHAT WAS
10786                                                                    ;ON DISK BEFORE WRITE WITH
10787                                                                    ;WRITE LOCK WAS ATTEPTED
10788                                                                    ;
10789                                                                    ;RECEIVED DATA GIVES WHAT
10790                                                                    ;WAS READ BACK AFTER WRITE
10791                                                                    ;WITH WRITE LOCKED WAS ATTEPTED
10792
10793                                  ;*SAVE REGISTERS FOR COMPARISON AFTER WRITE LOCK HAS BEEN
10794                                  ;*UNLOCKED
10795
10796  040402                  8$:
10797  040402  004037  041672                  JSR     R0,@#SAVER      ;SAVE REGISTERS
10798  040406  002272                          RHWC                    ;RHWC IS THE FIRST REGISTER SAVED
10799  040410  004612                          SAVERE                  ;STARTING ADDRESS OF WHERE
10800                                                                    ;THE REGISTERS ARE SAVED
10801  040412  000022                          18.                     ;NUMBER OF REGISTERS
10802                                                                    ;SAVED = 18.
10803
10804  040414                  ST20:
10805  040414  104401  040422                  TYPE    ,65$            ;;TYPE ASCIZ STRING
10806  040420  000407                          BR      64$             ;;GET OVER THE ASCIZ
10807                          ;;65$:  .ASCIZ  <15><12>/ON DRIVE  /
10808  040440                  64$:
10809  040440  013746  004716                  MOV     @#UNIT,-(SP)    ;GET UNIT UNDER TEST
10810  040444  104405                          TYPDS
10811  040446  104401  040454                  TYPE    ,67$            ;;TYPE ASCIZ STRING
10812  040452  000440                          BR      66$             ;;GET OVER THE ASCIZ
```

```
10813                                   ;;67$:  .ASCIZ  <15><12>/PUSH WRITE PROTCT BUTTON TO UNLOCK WRITES THEN HIT CONTINUE/<15
10814  040554                           66$:
10815  040554  000000                           HALT
10816
10817                                           ;*THE ONLY BIT THAT SHOULD CHANGE IS WRL-BIT #11 IN RHDS1
10818
10819  040556  004037  042404                   JSR     R0,@#CHREG      ;CHANGE BITS IN SAVED REGISTER
10820  040562  002322                           RHDS1                   ;CHANGE RHDS1 REGISTER
10821
10822  040564  000001                           1                       ;1 BIT/BITS TO BE CHANGED
10823  040566  000000                           0                       ;NEW VALUE OF WRL IS 0
10824  040570  004000                           WRL                     ;CHANGE WRL BIT
10825
10826                                           ;*COMPARE ALL REGISTERS BEFORE WRITE LOCK WAS UNLOCKED
10827                                           ;*WITH REGISTERS AFTER WRITE WAS UNLOCKED
10828
10829
10830  040572  004037  042512                   JSR     R0,@#COMREG     ;COMPARE SAVED REGISTERS WITH
10831                                                                   ;PRESENT VALUE
10832  040576  004612                           SAVERE                  ;GOOD DATA SAVED IN 'SAVERE'
10833  040600  002354                           WC                      ;TEST DATA STARTING FROM 'RHWC'
10834  040602  000022                           18.                     ;18. REGISTERS TO BE COMPARED
10835  040604  040610                           9$                      ;RETURN TO 9$ ON ERROR
10836  040606  040614                           10$                     ;RETURN TO 10$ ON NO ERROR
10837
10838  040610  104044                   9$:     ERROR   44              ;UNLOCKING WRITES BY WRITE
10839  040612  000207                           RTS     PC              ;LOCK BUTTON CAUSED AN ERROR
10840                                                                   ;GOOD DATA GIVES WHAT SHOULD
10841                                                                   ;BE THERE
10842                                                                   ;RECEIVED DATA GIVES WHAT WAS
10843                                                                   ;THERE AFTER WRITES WERE
10844                                                                   ;UNLOCKED
10845                                                                   ;ON THIS ERROR NO LOOPING IS RECOMMENDED
10846                                                                   ;JUST A HALT ON ERROR WILL DO THE SAME
10847                                                                   ;THING AS ONLY THE REGISTERS ARE READ
10848  040614  012737  177777  047342   10$:     MOV     #-1,@#PRITEM    ;CLEAR PREVIOUS ITEM NUMBER
```

```
10849
10850
10851                                   ;;*****************************************************************
10852                                   ;;*****************************************************************
10853                                   ;;*****************************************************************
10854                                   ;*TEST 51        END OF DRIVE
10855
10856                                   ;*      THIS IS THE END OF TEST FOR ONE DRIVE
10857
10858                                   ;*      IF THERE ARE MORE DRIVES, THEN THE PROGRAM
10859                                   ;*      JUMPS TO TEST 4 FOR TESTING THE NEXT DRIVE
10860
10861                                   ;*      END PASS IS REACHED ONLY AFTER ALL DRIVES ARE TESTED
10862
10863                                   ;;*****************************************************************
10864  040622  000004                   TST51:  SCOPE
10865  040624  012737  000001  001212           MOV     #1,$TIMES       ;;DO 1 ITERATION
10866  040632  012737  000000  177776           MOV     #0,PS           ;REINSTATE PS TO 0
10867
10868  040640  104401  040646                 TYPE    ,65$            ;;TYPE ASCIZ STRING
10869  040644  000424                   BR      64$             ;;GET OVER THE ASCIZ
10870                                   ;;65$:  .ASCIZ  <15><12>/TOTAL ERRORS ON THIS PASS ON UNIT NO./
10871  040716                           64$:
10872  040716  013746  004716                 MOV     @#UNIT,-(SP)    ;GET READY TO TYPE UNIT NUMBER
10873  040722  104405                   TYPDS
10874  040724  104401  040732                 TYPE    ,67$            ;;TYPE ASCIZ STRING
10875  040730  000402                   BR      66$             ;;GET OVER THE ASCIZ
10876                                   ;;67$:  .ASCIZ  / =/
10877  040736                           66$:
10878  040736  013746  001112                 MOV     @#$ERTTL,-(SP)  ;GET READY TO TYPE NUMBER OF ERRORS
10879  040742  104405                   TYPDS
10880  040744  005037  001112                 CLR     @#$ERTTL        ;CLEAR TOTAL NUMBER OF ERRORS
10881  040750  005037  001102                 CLR     @#$TSTNM        ;CLEAR TEST NUMBER
10882  040754  005737  004726                 TST     @#SELECT        ;STARTING FROM 200 ?
10883  040760  001413                   BEQ     3$              ;CHECK NEXT DRIVE IF SO
10884                                                           ;CONTINUE WITH THIS ONE IF NOT
10885
10886  040762  005237  001100                 INC     @#$PASS         ;INCREASE PASS COUNT
10887  040766  104401  041157                 TYPE    ,$ENDMG         ;TYPE "END PASS #"
10888  040772  013746  001100                 MOV     @#$PASS,-(SP)
10889  040776  104405                   TYPDS
10890  041000  104401  041154                 TYPE    ,$ENULL
10891  041004  000137  010102                 JMP     @#TST4          ;JUMP TEST 4 ---------------------->
10892
10893  041010  012737  177777  047342  3$:    MOV     #-1,@#PRITEM    ;CLEAR PREVIOUS ITEM NUMBER
10894  041016  005337  004720                 DEC     @#NOUNITS       ;NO. OF UNITS PRESENT
10895  041022  001413                   BEQ     $EOP            ;BRANCH IF ALL DRIVES COMPLETE
10896  041024  013700  004716                 MOV     @#UNIT,R0       ;UNIT UNDER TEST
10897  041030  012701  004676                 MOV     #UNITS,R1       ;TABLE POINTER
10898  041034  022100                   1$:     CMP     (R1)+,R0        ;IS THIS UNIT JUST TESTED ?
10899  041036  001401                   BEQ     2$              ;BRANCH IF YES
10900  041040  000775                   BR      1$              ;BRANCH IF NO
10901  041042  011137  004716           2$:     MOV     (R1),@#UNIT     ;MAKE THIS NEXT UNIT
10902  041046  000137  010102                 JMP     @#TST4          ;TEST THE NEXT DRIVE -------------->
10903
```

```
10904                                   .SBTTL
10905                                   .SBTTL  **SUBROUTINES**
10906                                   .SBTTL
10907
10908
10909                           .SBTTL  END OF PASS ROUTINE
10910
10911                           ;;**********************************************************************
10912                           ;*INCREMENT THE PASS NUMBER ($PASS)
10913                           ;*TYPE "END PASS #XXXXX" (WHERE XXXXX IS A DECIMAL NUMBER)
10914                           ;*IF THERES A MONITOR GO TO IT
10915                           ;*IF THERE ISN'T JUMP TO TST1
10916
10917  041052                   $EOP:
10918  041052  000004                   SCOPE
10919  041054  005037  001102           CLR     $TSTNM          ;;ZERO THE TEST NUMBER
10920  041060  005037  001212           CLR     $TIMES          ;;ZERO THE NUMBER OF ITERATIONS
10921  041064  005237  001100           INC     $PASS           ;;INCREMENT THE PASS NUMBER
10922  041070  042737  100000  001100   BIC     #100000,$PASS   ;;DON'T ALLOW A NEG. NUMBER
10923  041076  005327                   DEC     (PC)+           ;;LOOP?
10924  041100  000001           $EOPCT: .WORD   1
10925  041102  003022                   BGT     $DOAGN          ;;YES
10926  041104  012737                   MOV     (PC)+,@(PC)+    ;;RESTORE COUNTER
10927  041106  000001           $ENDCT: .WORD   1
10928  041110  041100                   $EOPCT
10929  041112  104401  041157           TYPE    ,$ENDMG         ;;TYPE "END PASS #"
10930  041116  013746  001100           MOV     $PASS,-(SP)     ;;SAVE $PASS FOR TYPEOUT
10931  041122  104405                   TYPDS                   ;;GO TYPE--DECIMAL ASCII WITH SIGN
10932  041124  104401  041154           TYPE    ,$ENULL         ;;TYPE A NULL CHARACTER
10933  041130  013700  000042   $GET42: MOV     @#42,R0         ;;GET MONITOR ADDRESS
10934  041134  001405                   BEQ     $DOAGN          ;;BRANCH IF NO MONITOR
10935  041136  000005                   RESET                   ;;CLEAR THE WORLD
10936  041140  004710           $ENDAD: JSR     PC,(R0)         ;;GO TO MONITOR
10937  041142  000240                   NOP                     ;;SAVE ROOM
10938  041144  000240                   NOP                     ;;FOR
10939  041146  000240                   NOP                     ;;ACT11
10940  041150                   $DOAGN:
10941  041150  000137                   JMP     @(PC)+          ;;RETURN
10942  041152  006314           $RTNAD: .WORD   TST1
10943  041154     377     377     000   $ENULL: .BYTE   -1,-1,0         ;;NULL CHARACTER STRING
10944  041157     015  042412  042116   $ENDMG: .ASCIZ  <15><12>/END PASS #/
10945  041164  050040  051501  020123
10946  041172  000043
10947
```

```
10948
10949
10950                                           .SBTTL  JAM CURRENT CYLINDER ROUTINE
10951
10952                                           ;*THIS ROUTINE WILL CHANGE THE CURRENT CYLINDER REGISTER - 'RHCC'
10953                                           ;*BY GIVING A 'SEEK' COMMAND FOLLOWED BY AN INIT WHICH WILL LOAD
10954                                           ;*'RHCC' WITH THE DESIRED CYLINDER VALUE.  THE ROUTINE THEN CHECKS
10955                                           ;*THAT THE LOADED VALUE IS CORRECT.
10956                                           ;*
10957                                           ;*CALL IS:
10958                                           ;*      JSR     R0,@#MAKECYL
10959                                           ;*      XC                              ;DESIRED VALUE OF CURRENT CYLINDER
10960
10961  041174                                   MAKECYL:
10962  041174  010546                                  MOV     R5,-(SP)                ;;PUSH R5 ON STACK
10963  041176  010037  041560                          MOV     R0,@#PCJSR              ;PC OF JSR+4
10964  041202  162737  000004  041560                  SUB     #4,@#PCJSR              ;SAVE PC OF JSR
10965  041210  012005                                  MOV     (R0)+,R5                ;GETTING READY TO FILL DESIRED CYLINDER
10966  041212  010577  141074                          MOV     R5,@RHCA                ;FILL DESIRED CYLINDER REGISTER
10967  041216  005077  141062                          CLR     @RHDST                  ;MAKE SURE DESIRED SECTOR TRACK IS NOT ILLEGAL
10968  041222  013777  002452  141050                  MOV     @#SEECOM,@RHCS1         ;FILL SEEK COMMAND
10969  041230  012777  000001  141062                  MOV     #DMD,@RHMR              ;SET DIAGNOSTIC MODE
10970
10971  041236  004737  043436                          JSR     PC,@#PUTREG             ;TAKE A REGISTER SNAPSHOT
10972  041242  032737  040000  002404                  BIT     #ERR,@#DS1              ;CHECK FOR COMPOSITE ERROR
10973  041250  001401                                  BEQ     2$                      ;NOT = 1, A-OK
10974  041252  104103                                  ERROR   103                     ;REGISTER CONTENTS INCORRECT BEFORE A
10975                                                                                  ;DIAGNOSTIC SEEK
10976
10977  041254  052777  000001  141016  2$:             BIS     #GO,@RHCS1              ;ISSUE 'GO' TO SEEK COMMAND
10978  041262  000240                                  NOP                             ;ALLOW TIME FOR SEEK TO HANG UP
10979  041264  000240                                  NOP                             ;ALLOW TIME FOR SEEK TO HANG UP
10980  041266  000240                                  NOP                             ;ALLOW TIME FOR SEEK TO HANG UP
10981  041270  000240                                  NOP                             ;ALLOW TIME FOR SEEK TO HANG UP
10982
10983  041272  004737  043436                          JSR     PC,@#PUTREG             ;TAKE A 2ND REGISTER SNAPSHOT
10984  041276  032737  040000  002404                  BIT     #ERR,@#DS1              ;CHECK FOR ERRORS
10985  041304  001401                                  BEQ     3$                      ;NOT = 1, A-OK
10986  041306  104104                                  ERROR   104                     ;REGISTER CONTENTS INCORRECT AFTER
10987                                                                                  ;A DIAGNOSTIC SEEK
10988
10989  041310  004737  041524          3$:             JSR     PC,@#CLDISK             ;GIVE INIT TO FORCE THE TRANSFER
10990  041314  017737  141014  001126                  MOV     @RHCC,@#$BDDAT          ;TEST DATA
10991  041322  020537  001126                          CMP     R5,@#$BDDAT             ;COMPARE CURRENT CYLINDER
10992  041326  001406                                  BEQ     1$                      ;BRANCH IF GOOD
10993  041330  010537  001124                          MOV     R5,@#$GDDAT             ;GOOD VALUE OF RHCC
10994  041334  013737  002334  004600                  MOV     @#RHCC,@#REGADR         ;FAILING REGISTER ADDRESS
10995  041342  104077                                  ERROR   77                      ;CURRENT CYLINDER DOES NOT MATCH DESIRED CYLINDER
10996                                                                                  ;REGISTER AFTER A SEEK AND AN INIT
10997  041344                                   1$:
10998  041344  012605                                  MOV     (SP)+,R5                ;;POP STACK INTO R5
10999  041346  000200                                  RTS     R0
```

```
11000
11001                                   ;*THIS FILLS MEMORY WITH GIVEN DATA
11002                                   ;*USED CHIEFLY FOR HEADER INFORMATION
11003                                   ;*CALL IS
11004                                   ;*      JSR     R0,@#FLHEAD     ;FILL HEADER
11005                                   ;*      LOC                     ;LOCATION WHERE SAVED
11006                                   ;*      XN                      ;NUMBER OF WORDS
11007                                   ;*      XD1                     ;DATA REPEATED XN TIMES
11008                                   ;*      XD2                     ;DATA REPEATED XN TIMES
11009                                   ;*
11010                                   ;*
11011                                   ;*
11012
11013  041350                           FLHEAD:
11014  041350  010146                           MOV     R1,-(SP)        ;;PUSH R1 ON STACK
11015  041352  010246                           MOV     R2,-(SP)        ;;PUSH R2 ON STACK
11016  041354  012001                           MOV     (R0)+,R1        ;R1 HAS ADDRESS OF WHERE TO SAVE
11017  041356  012002                           MOV     (R0)+,R2        ;R2 HAS NUMBER OF WORDS
11018
11019                                           ;*NOW FILL DATA
11020
11021  041360  012021                   1$:     MOV     (R0)+,(R1)+     ;SAVE DATA
11022  041362  005302                           DEC     R2              ;DECREMENT COUNT
11023  041364  001375                           BNE     1$              ;BRANCH IF INCOMPLETE
11024  041366  012602                           MOV     (SP)+,R2        ;;POP STACK INTO R2
11025  041370  012601                           MOV     (SP)+,R1        ;;POP STACK INTO R1
11026  041372  000200                           RTS     R0
11027
11028
11029
11030                                   ;*THIS CLEARS ANY BLOCK OF MEMORY.
11031                                   ;*FILLING IT WITH ANY DATA
11032                                   ;*CALL IS
11033                                   ;*      JSR     R0,@#CLAREA
11034                                   ;*      F                       ;FROM
11035                                   ;*      N                       ;NUMBER OF WORDS
11036                                   ;*      D                       ;DATA  TO BE FILLED
11037                                   ;*
11038                                   ;*R1 WILL HAVE STARTING ADDRESS OF BLOCK TO BE FILLED
11039                                   ;*R2 WILL HAVE NUMBER OF WORDS
11040                                   ;*R3 WILL HAVE DATA
11041
11042  041374                           CLAREA:
11043  041374  010146                           MOV     R1,-(SP)        ;;PUSH R1 ON STACK
11044  041376  010246                           MOV     R2,-(SP)        ;;PUSH R2 ON STACK
11045  041400  010346                           MOV     R3,-(SP)        ;;PUSH R3 ON STACK
11046  041402  012001                           MOV     (R0)+,R1        ;FROM
11047  041404  012002                           MOV     (R0)+,R2        ;NUMBER
11048  041406  012003                           MOV     (R0)+,R3        ;DATA
11049  041410  010321                   1$:     MOV     R3,(R1)+        ;MOVE DATA
11050  041412  005302                           DEC     R2              ;COUNT
11051  041414  001375                           BNE     1$              ;BRANCH IF NOT COMPLETE
11052  041416  012603                           MOV     (SP)+,R3        ;;POP STACK INTO R3
11053  041420  012602                           MOV     (SP)+,R2        ;;POP STACK INTO R2
11054  041422  012601                           MOV     (SP)+,R1        ;;POP STACK INTO R1
11055  041424  000200                           RTS     R0              ;RETURN TO MAIN PROGRAM
```

```
11056
11057
11058
11059
11060
11061
11062                                   ;*THIS IS A SUBROUTINE TO FILL SAVED REGISTER LOCATION
11063                                   ;*WITH GIVEN VALUE
11064                                   ;*CALL IS
11065                                   ;*      JSR     R0,@#FILLRE
11066                                   ;*      RHXX            ;REGISTER NAME
11067                                   ;*      D               ;DATA
11068                                   ;*
11069
11070  041426                           FILLRE:
11071  041426  010146                          MOV     R1,-(SP)        ;;PUSH R1 ON STACK
11072  041430  010246                          MOV     R2,-(SP)        ;;PUSH R2 ON STACK
11073  041432  012001                          MOV     (R0)+,R1        ;ADDRESS OF ADDRESS OF REGISTER
11074  041434  012002                          MOV     (R0)+,R2        ;DATA
11075  041436  162701  002272                  SUB     #RHWC,R1        ;OFSET
11076  041442  010261  004612                  MOV     R2,SAVERE(R1)   ;DATA IS MOVED IN
11077  041446  012602                          MOV     (SP)+,R2        ;;POP STACK INTO R2
11078  041450  012601                          MOV     (SP)+,R1        ;;POP STACK INTO R1
11079  041452  000200                          RTS     R0              ;RETURN TO MAIN PROGRAM
11080
11081
11082
```

```
11083                                   ;*THIS SUBROUTINE SETS UP FOR SEARCH
11084                                   ;*CALL IS
11085                                   ;*      JSR     R0,@#SRCH
11086                                   ;*      C               ;CYLINDER
11087                                   ;*.BYTE S               ;SECTOR
11088                                   ;*.BYTE T               ;TRACK
11089
11090  041454  012077  140632          SRCH:   MOV     (R0)+,@RHCA     ;SET DESIRED CYLINDER ADDRESS
11091  041460  012077  140620                  MOV     (R0)+,@RHDST    ;SET DESIRED SECTOR/TRACK ADDRESS
11092  041464  013777  002434  140606          MOV     @#SERCH,@RHCS1  ;GET READY FOR SEARCH
11093                                                                  ;WITH 30 IN RHCS1
11094  041472  000200                         RTS     R0
11095
11096
11097
11098
11099
11100
11101
11102                                   ;*THIS SUBROUTINE SETS UP FOR SEEK COMMANDS
11103                                   ;*CALL IS
11104                                   ;*      JSR     R0,@#SEEKCY
11105                                   ;*      C               ;CYLINDER
11106                                   ;*
11107
11108  041474  012077  140612          SEEKCY: MOV     (R0)+,@RHCA     ;SET DESIRED CYLINDER ADDRESS
11109  041500  013777  002452  140572          MOV     @#SEECOM,@RHCS1 ;MOV 4 INTO RHCS1
11110  041506  000200                         RTS     R0              ;RETURN TO MAIN PROGRAM
```

```
11111
11112                                   ;*THIS SUBROUTINE SETS UP FOR OFFSET COMMANDS
11113                                   ;*CALL IS
11114                                   ;*      JSR     R0,@#OFSET
11115                                   ;*      0               ;MICRO INCHES OFSET
11116
11117  041510  052077  140574          OFSET:  BIS     (R0)+,@RHOF     ;SET OFSET REGISTER
11118  041514  013777  002454  140556          MOV     @#OFSETC,@RHCS1 ;MOV14 INTO RHCS1
11119  041522  000200                          RTS     R0              ;RETURN TO MAIN PROGRAM
11120
11121
11122  041524  013701  002300          CLDISK: MOV     @#RHCS1,        R1      ;R1 WILL BE CONTROL AND STATUS1
11123  041530  013702  002276                  MOV     @#RHCS2,        R2      ;R2 WILL BE CONTROL AND STATUS2
11124  041534  013703  002322                  MOV     @#RHDS1,        R3      ;R3 WILL BE DISK STATUS REGISTER1
11125  041540  013704  002302                  MOV     @#RHER1,        R4      ;R4 WILL BE ERROR REGISTER #1
11126
11127  041544  012712  000040                  MOV     #CLR,@R2                ;CLEAR ALL REG.
11128  041550  013712  004716                  MOV     @#UNIT,@R2      ;REINSTATE UNIT NO.
11129  041554  005011                          CLR     @R1                     ;CLEAR FUNCTION BITS
11130  041556  000207                          RTS     PC
```

```
11131
11132
11133
11134
11135                                   ;*THIS CHECKS THAT DEVICE AVAILABLE (DVA) AND READY (RDY) IN RHCS1 = 1
11136                                   ;*AND CHECKS MEDIUM ON LINE (MOL), DEVICE PRESENT (DPR), DEVICE READY
11137                                   ;* (DRY) IN RHDS1 = 1
11138
11139                                   ;*IT ALSO CHECKS THAT THERE ARE NO BITS STUCK AT 1 IN RHDS1
11140
11141
11142  041560  000000                   PCJSR:  0                       ;PC OF JSR
11143
11144  041562  011637  041560           CHECK:  MOV     (SP),@#PCJSR    ;SAVE PC OF JSR+4
11145  041566  162737  000004  041560           SUB     #4,@#PCJSR      ;GET PC OF JSR
11146  041574  011346                           MOV     @R3,-(SP)       ;GET RHDS1
11147  041576  052716  000100                   BIS     #VV,(SP)        ;DONT CHECK VV BIT
11148  041602  000406                           BR      CHECKC          ;GOTO COMMON CHECK ROUTINE
11149
11150  041604  011637  041560           CHECKT: MOV     (SP),@#PCJSR    ;SAVE PC OF JSR+4
11151  041610  162737  000004  041560           SUB     #4,@#PCJSR      ;GET PC OF JSR
11152  041616  011346                           MOV     @R3,-(SP)       ;GET RHDS1 & DO VV CHECK AT 3$
11153
11154  041620  011146                   CHECKC: MOV     @R1,-(SP)       ;GET CS1
11155  041622  042716  173577                   BIC     #173577,(SP)    ;CLEAR UNWANTED BITS
11156  041626  022726  004200                   CMP     #DVA!RDY,(SP)+  ;RHCS1 SHOULD HAVE DEVICE AVAILABLE
11157                                                                   ;AND BE READY
11158  041632  001403                           BEQ     3$              ;BRANCH IF IT DOES
11159  041634  011137  001122                   MOV     @R1,@#$BDADR    ;BAD DATA REGISTER (RHCS1)
11160  041640  104061                           ERROR   61              ;RHCS1 DID NOT HAVE DEVICE
11161                                                                   ;AVAILABLE RIGHT AT THE START
11162                                                                   ;ALL OTHER BITS SHOULD BE 0
11163
11164  041642  042716  102000           3$:     BIC     #ATA!LBT,(SP)   ;CLEAR UNWANTED BITS
11165  041646  022726  010700                   CMP     #MOL!DPR!DRY!VV,(SP)+ ;RHDS1 SHOULD HAVE THESE SET
11166  041652  001404                           BEQ     7$              ;BRANCH IF GOOD
11167  041654  011337  001122                   MOV     @R3,@#$BDADR    ;BAD DATA IN  REGISTER (RHDS1)
11168  041660  104062                           ERROR   62              ;RHDS1 HAS SOME BITS OTHER
11169                                                                   ;THAN MOL, DRY, DPR,VV SET
11170                                                                   ;ALL OTHER BITS SHOULD BE 0
11171  041662  000207                           RTS     PC              ;RETURN TO TEST AND HALT/CONTINUE
11172                                                                   ;DEPENDING ON WHETHER THIS IS A
11173                                                                   ;"FATAL" ERROR
11174
11175  041664  062716  000006           7$:     ADD     #6,(SP)         ;ADJUST STACK TO JUMP OVER HALT IN TEST
11176  041670  000207                           RTS     PC              ;RETURN TO THE TEST AND CONTINUE
11177
```

```
11178
11179                                   ;*THIS IS A SUBROUTINE TO SAVE REGISTERS
11180                                   ;*IN THE REGISTER TABLE TO ANY LOCATION
11181                                   ;*THE CALL IS
11182                                   ;*JSR   R0,@#SAVER
11183                                ;*      F       ;FROM
11184                                ;*      T       ;TO
11185                                ;*      N       ;NUMBER OF WORDS SAVED
11186                                ;*F MUST ALWAYS BE RHCS1
11187                                ;*T MUST ALWAYS BE SAVRE
11188
11189  041672                        SAVER:
11190  041672  010146                       MOV     R1,-(SP)        ;;PUSH R1 ON STACK
11191  041674  010246                       MOV     R2,-(SP)        ;;PUSH R2 ON STACK
11192  041676  010346                       MOV     R3,-(SP)        ;;PUSH R3 ON STACK
11193  041700  012001                       MOV     (R0)+,R1        ;FROM
11194  041702  012002                       MOV     (R0)+,R2        ;TO
11195  041704  012003                       MOV     (R0)+,R3        ;NUMBER
11196  041706  013122                1$:    MOV     @(R1)+,(R2)+    ;SAVE REGISTER CONTENTS
11197  041710  005303                       DEC     R3              ;COUNT
11198  041712  001375                       BNE     1$              ;BRANCH IF NOT DONE
11199  041714  012603                       MOV     (SP)+,R3        ;;POP STACK INTO R3
11200  041716  012602                       MOV     (SP)+,R2        ;;POP STACK INTO R2
11201  041720  012601                       MOV     (SP)+,R1        ;;POP STACK INTO R1
11202  041722  000200                       RTS     R0
11203
11204
11205
11206
11207
11208
11209
11210
11211
11212
11213
11214
11215
11216
11217
11218                                   ;*WHEN AN EVENT IS TO BE TIMED THE RP04 VECTORS TO "TIME 1"
11219                                   ;*PRIORITY OF PROCESS OR IS 4
11220                                   ;*PRIORITY OF TRAPS MUST BE 6
11221                                   ;*PRIORITY OF RP04 INTERRUPTS IS 7
11222                                   ;*
11223
11224  041724  005077  140414        TIME1:  CLR     @PCLCSR                 ;STOP THE CLOCK
11225  041730  017737  140414  041762        MOV     @PCLCTR,@#WAITTM ;GET TIME ON CLOCK
11226  041736  017737  140372  004664  TIME2: MOV     @RHCC,@#FINACC          ;GET CURRENT CYLINDER
11227  041744  017737  140366  004662        MOV     @RHLA,@#FINALA          ;GET LOOK AHEAD
11228  041752  000002                       RTI                     ;RETURN TO WAIT P OR WAIT.T
11229
11230
11231
11232                                   ;*THIS IS A WAIT LOOP WHEN AN EVENT IS TO BE TIMED
11233                                   ;*THE CALL IS
```

```
11234                                   ;*      WAT
11235                                   ;*      A       ;ABSOLUTE REGISTER ADDRESS
11236                                   ;*      B       ;BIT WAITED FOR
11237                                   ;*      TA      ;TIME ALLOWED GIVEN IN 10 MICROSEC
11238                                   ;*      TO      ;TOLERANCE PLUS/MINUS IN 10 MICROSEC
11239                                   ;*
11240                                   ;*R1-WILL HAVE TIME ALLOWED IN 10 MICRO SECONDS
11241                                   ;*R2-WILL HAVE TOLERANCE PLUS/MINUS IN 10 MICRO SECONDS
11242                                   ;*MINIMUM TIME THAT CAN BE MEASURED IS ABOUT 12 MICRO SECONDS
11243                                   ;*FOR THE SLOWEST PROCESSOR
11244
11245  041754  000000                   WAITPC: 0                       ;WAT PC
11246  041756  000000                   WAITRE: 0                       ;WAIT ON REGISTER ADDRESS
11247  041760  000000                   WAITBT: 0                       ;WAIT ON BIT
11248  041762  000000                   WAITTM: 0                       ;WAITED TIME
11249  041764  005037  041762           WAIT.P: CLR     @#WAITTM        ;CLEAR WAITED TIME
11250  041770  005077  140352                   CLR     @PCLBUF ;CLEAR COUNT SET BUFFER
11251  041774  012777  000021  140342           MOV     #GO!BIT4,@PCLCSR ;COUNT UP, 100 KHZ, START CLOCK
11252  042002  010046                           MOV     R0,-(SP)        ;;PUSH R0 ON STACK
11253  042004  010146                           MOV     R1,-(SP)        ;;PUSH R1 ON STACK
11254  042006  010246                           MOV     R2,-(SP)        ;;PUSH R2 ON STACK
11255  042010  010346                           MOV     R3,-(SP)        ;;PUSH R3 ON STACK
11256  042012  016600  000010                   MOV     10(SP),R0       ;R0 HAS ADDRESS OF NEXT LOCATION
11257  042016  010037  041754                   MOV     R0,@#WAITPC     ;NOW WAITPC HAS WAT PC + 2
11258  042022  162737  000002  041754           SUB     #2,@#WAITPC     ;WAT PC IS IN WAITPC
11259  042030  013037  041756                   MOV     @(R0)+,@#WAITRE ;WAIT ON REGISTER ADDRESS
11260  042034  012037  041760                   MOV     (R0)+,@#WAITBT  ;WAIT ON BIT
11261  042040  012001                           MOV     (R0)+,R1        ;R1 HAS TIME IN 10 MSEC
11262  042042  012002                           MOV     (R0)+,R2        ;R2 HAS TOLERANCE IN 10 MSEC
11263  042044  010066  000010                   MOV     R0,10(SP)       ;RESTORE RETURN ON STACK
11264
11265                                           ;*THIS SECTION WAITS FOR BIT, THROUGH TWO COUNT DOWNS
11266  042050  013703  042222                   MOV     @#TIMCNT,R3     ;R3 IS A TEMPORARY COUNTER
11267  042054  033777  041760  177674   1$:     BIT     @#WAITBT,@WAITRE ;IS REQUIRED BIT THERE
11268  042062  001025                           BNE     4$              ;BRANCH IF YES
11269  042064  005303                           DEC     R3              ;COUNT IF REQUIRED BIT NOT THERE
11270  042066  001372                           BNE     1$
11271  042070  013703  042222                   MOV     @#TIMCNT,R3     ;TEMPORARY COUNTER
11272  042074  033777  041760  177654   2$:     BIT     @#WAITBT,@WAITRE ;IS REQUIRED BIT THERE
11273  042102  001015                           BNE     4$              ;BRANCH IF YES
11274  042104  005303                           DEC     R3              ;COUNT IF REQUIRED BIT NOT THERE
11275  042106  001372                           BNE     2$
11276  042110  017737  177642  001126           MOV     @WAITRE,@#$BDDAT ;REGISTER CONTENTS FOR TYPEOUT
11277  042116  032777  000100  140154           BIT     #IE,@RHCS1      ;DID ANY INTERRUPT OCCUR
11278  042124  001402                           BEQ     3$              ;BRANCH IF YES
11279  042126  104001                           ERROR   1               ;RP04 DID NOT INTERRUPT
11280  042130  000427                           BR      7$              ;OUT
11281  042132  104002                   3$:     ERROR   2               ;RP04 INTERRUPTED BUT WAITED
11282                                                                   ;ON BIT DID NOT OCCUR
11283                                                                   ;EVEN AFTER TWO COUNT DOWNS
11284                                                                   ;FROM 177777 TO 0
11285  042134  000425                           BR      7$              ;OUT
11286
11287                                           ;*NOW TIME AND TOLERANCE WILL BE CHECKED
11288  042136  017737  177614  001126   4$:     MOV     @WAITRE,@#$BDDAT ;REGISTER CONTENTS FOR TYPEOUT
11289  042144  032777  000100  140126           BIT     #IE,@RHCS1      ;DID ANY INTERRUPT OCCUR
```

```
11290  042152  001402                    BEQ     5$              ;BRANCH IF YES
11291  042154  104003                    ERROR   3               ;INTERRUPT DID NOT OCCUR EVEN
11292                                                            ;AFTER ONE BNE AND ONE MOV
11293                                                            ;OF THE WAITED ON BIT SETTING
11294  042156  000414                    BR      7$              ;OUT
11295  042160  160201            5$:     SUB     R2,R1           ;R1 NOW HAS LOWER LIMIT OF TIME
11296  042162  023701  041762            CMP     @#WAITTM,R1     ;FOR GOOD RESULTS, WAITTM
11297                                                            ;MUST BE GREATER OR EQUAL
11298                                                            ;TOR1
11299  042166  103002                    BHIS    6$              ;BRANCH IF GOOD
11300  042170  104004                    ERROR   4               ;BIT DID OCCUR BUT TIME
11301                                                            ;TAKEN IS BELOW LOWER LIMIT
11302  042172  000406                    BR      7$              ;OUT
11303
11304  042174  060202            6$:     ADD     R2,R2           ;DOUBLE TOLERANCE
11305  042176  060201                    ADD     R2,R1           ;R1 NOW HAS UPPER LIMIT OF TIME
11306  042200  020137  041762            CMP     R1,@#WAITTM     ;FOR GOOD RESULTS, WAITTM
11307                                                            ;MUST BE LESS OR EQUAL TO R1
11308  042204  103001                    BHIS    7$              ;BRANCH IF GOOD
11309  042206  104004                    ERROR   4               ;BIT DID OCCUR BUT TIME TAKEN
11310                                                            ;IS ABOVE UPPER LIMIT
11311  042210            7$:
11312  042210  012603                    MOV     (SP)+,R3        ;;POP STACK INTO R3
11313  042212  012602                    MOV     (SP)+,R2        ;;POP STACK INTO R2
11314  042214  012601                    MOV     (SP)+,R1        ;;POP STACK INTO R1
11315  042216  012600                    MOV     (SP)+,R0        ;;POP STACK INTO R0
11316  042220  000002                    RTI                     ;RETURN TO MAIN TEST
11317
11318
11319
11320
11321
11322
11323                            ;*THIS IS A WAIT LOOP WHEN NO P-CLOCK IS AVAILABLE
11324                            ;*NO TIMING IS DONE
11325                            ;*CALL IS
11326                            ;*      WAT
11327                            ;*      A       ;ABSOLUTE REGISTER ADDRESS
11328                            ;*      B       ;BIT WAITE) FOR
11329                            ;*      TA      ;TIME-NOT USED HERE
11330                            ;*      TO      ;TIME-NOT USED HERE
11331                            ;*R3-IS A TEMPORARY COUNTER
11332  042222  177777            TIMCNT: 177777                  ;COUNT FOR WAIT LOOP
11333
11334  042224                    WAIT.T:
11335  042224  010046                    MOV     R0,-(SP)        ;;PUSH R0 ON STACK
11336  042226  010346                    MOV     R3,-(SP)        ;;PUSH R3 ON STACK
11337  042230  016600  000004            MOV     4(SP),R0        ;R0 HAS ADDRESS OF NEXT LOCATION
11338  042234  010037  041754            MOV     R0,@#WAITPC     ;WAT PC +2 IS IN WAITPC
11339  042240  162737  000002  041754    SUB     #2,@#WAITPC     ;WAT PC IS IN WAITPC
11340  042246  013037  041756            MOV     @(R0)+,@#WAITRE ;WAIT ON REGISTER ADDRESS
11341  042252  012037  041760            MOV     (R0)+,@#WAITBT  ;WAIT ON BIT
11342  042256  022020                    CMP     (R0)+,(R0)+     ;DUMP NEXT TWO WORDS-TA, TO
11343  042260  010066  000004            MOV     R0,4(SP)        ;RESTORE RETURN ON STACK
11344
11345                                    ;*THIS HAS THE TWO COUNT DOWNS FROM 177777
```

```
11346  042264  013703  042222                  MOV     @#TIMCNT,R3     ;R3 HAS TEMPORARY COUNT
11347  042270  033777  041760  177460  1$:     BIT     @#WAITBT,@WAITRE ;IS REQUIRED BIT THERE
11348  042276  001025                          BNE     4$              ;BRANCH IF YES
11349  042300  005303                          DEC     R3              ;COUNT IF REQUIRED BIT NOT THERE
11350  042302  001372                          BNE     1$
11351  042304  013703  042222                  MOV     @#TIMCNT,R3     ;SECOND COUNT DOWN FROM 177777
11352  042310  033777  041760  177440  2$:     BIT     @#WAITBT,@WAITRE ;IS REQUIRED BIT THERE
11353  042316  001015                          BNE     4$              ;BRANCH IF YES
11354  042320  005303                          DEC     R3              ;COUNT IF REQUIRED BIT NOOT THERE
11355  042322  001372                          BNE     2$
11356  042324  017737  177426  001126          MOV     @WAITRE,@#$BDDAT ;REGISTER CONTENTS FOR TYPEOUT
11357  042332  032777  000100  137740          BIT     #IE,@RHCS1      ;DID ANY INTERRUPT OCCUR
11358  042340  001402                          BEQ     3$              ;BRANCH IF YES
11359  042342  104001                          ERROR   1               ;RP04 DID NOT INTERRUPT
11360                                                                  ;BIT DID NOT OCCUR
11361  042344  000414                          BR      5$              ;OUT
11362  042346  104002                  3$:     ERROR   2               ;RP04 INTERRUPTED BUT
11363                                                                  ;WAITED ON BIT DID NOT OCCUR
11364                                                                  ;EVEN AFTER TWO COUNT DOWNS
11365                                                                  ;FROM 177777 TO 0
11366  042350  000412                          BR      5$              ;OUT
11367
11368                                          ;*BIT DID SET SO CHECK IF INTERRUPT OCCURED
11369  042352  000240                  4$:     NOP                     ;ALLOW TIME FOR INTERRUPT
11370  042354  032777  000100  137716          BIT     #IE,@RHCS1      ;DID ANY INTERRUPT OCCUR
11371  042362  001405                          BEQ     5$              ;BRANCH IF YES
11372  042364  017737  177366  001126          MOV     @WAITRE,@#$BDDAT ;REGISTER CONTENTS FOR TYPEOUT
11373  042372  104003                          ERROR   3               ;INTERRUPT DID NOT OCCUR
11374                                                                  ;EVEN AFTER ONE BNE OF
11375                                                                  ;THE WAITED ON BIT OCCURING
11376  042374  000400                          BR      5$              ;OUT
11377  042376                          5$:
11378  042376  012603                          MOV     (SP)+,R3        ;;POP STACK INTO R3
11379  042400  012600                          MOV     (SP)+,R0        ;;POP STACK INTO R0
11380  042402  000002                          RTI                     ;RETURN TO MAIN TEST
```

```
11381                                   ;*THIS CHANGES REGISTER SAVED VALUE
11382                                   ;*CALL IS
11383                                   ;*      JSR     R0,@#CHREG
11384                                   ;*      R                       ;REGISTER TO BE CHANGED
11385                                   ;*      N                       ;NUMBER OF BITS TO BE CHANGED
11386                                   ;*      NEW                     ;NEW VALUE OF BIT MUST BE 0 OR1
11387                                   ;*      P                       ;POSITION OF BIT TO BE CHANGED
11388                                   ;*NEW AND P WILL BE REPEATED N NUMBER OF TIMES
11389  042404                           CHREG:
11390  042404  010146                          MOV     R1,-(SP)        ;;PUSH R1 ON STACK
11391  042406  010246                          MOV     R2,-(SP)        ;;PUSH R2 ON STACK
11392  042410  012001                          MOV     (R0)+,R1        ;R1 HAS ADDRESS OF ADDRESS OF REGISTER
11393  042412  012002                          MOV     (R0)+,R2        ;R2 HAS NUMBER OF CHANGES
11394  042414  162701  002272                 SUB     #RHWC,R1        ;R1 HAS OFSET OF REQUIRED REGISTER
11395  042420  005720                  1$:     TST     (R0)+           ;IS A BIC OR A BIS TO BE DONE
11396  042422  001403                          BEQ     2$              ;BRANCH IF A BIC IS REQUIRED
11397  042424  052061  004612                 BIS     (R0)+,SAVERE(R1) ;SET REQUIRED BIT
11398  042430  000402                          BR      3$              ;BRANCH TO DECREMENT COUNT
11399  042432  042061  004612          2$:     BIC     (R0)+,SAVERE(R1) ;CLEAR REQUIRED BIT
11400  042436  005302                  3$:     DEC     R2              ;DECREMENT NUMBER OF CHANGES
11401  042440  001367                          BNE     1$              ;BRANCH IF NOT COMPLETE
11402  042442  012602                          MOV     (SP)+,R2        ;;POP STACK INTO R2
11403  042444  012601                          MOV     (SP)+,R1        ;;POP STACK INTO R1
11404  042446  000200                          RTS     R0              ;RETURN TO MAIN PROGRAM
11405
11406
11407
11408
11409
11410
11411                                   ;*THIS FILLS A BLOCK WITH INCREMENTAL DATA
11412                                   ;*CALL IS
11413                                   ;*      JSR     R0,@#FILL
11414                                   ;*      F                       ;FROM
11415                                   ;*      N                       ;NUMBER OF WORDS
11416                                   ;*      S                       ;STARTING VALUE OF DATA
11417                                   ;*      I                       ;INCREMENT DATA BY
11418
11419  042450                           FILL:
11420  042450  010146                          MOV     R1,-(SP)        ;;PUSH R1 ON STACK
11421  042452  010246                          MOV     R2,-(SP)        ;;PUSH R2 ON STACK
11422  042454  010346                          MOV     R3,-(SP)        ;;PUSH R3 ON STACK
11423  042456  010446                          MOV     R4,-(SP)        ;;PUSH R4 ON STACK
11424  042460  012001                          MOV     (R0)+,R1        ;R1 HAS ADDRESS WHERE DATA IS TO GO
11425  042462  012002                          MOV     (R0)+,R2        ;R2 HAS NUMBER OF WORDS TO BE FILLED
11426  042464  012003                          MOV     (R0)+,R3        ;STARTING VALUE OF DATA
11427  042466  012004                          MOV     (R0)+,R4        ;R4 HAS INCREMENT
11428                                   ;*NOW DATA WILL BE FILLED
11429  042470  010321                  1$:     MOV     R3,(R1)+        ;FILL DATA
11430  042472  060403                          ADD     R4,R3           ;GET NEXT VALUE OF DATA
11431  042474  005302                          DEC     R2              ;DECREMENT COUNT
11432  042476  001374                          BNE     1$              ;BRANCH IF ALL NOT DONE
11433  042500  012604                          MOV     (SP)+,R4        ;;POP STACK INTO R4
11434  042502  012603                          MOV     (SP)+,R3        ;;POP STACK INTO R3
11435  042504  012602                          MOV     (SP)+,R2        ;;POP STACK INTO R2
11436  042506  012601                          MOV     (SP)+,R1        ;;POP STACK INTO R1
```

```
11437  042510  000200                    RTS     R0              ;RETURN TO MAIN PROGRAM
11438
11439
11440
```

```
11441
11442
11443
11444                                   ;*THIS IS A SUBROUTINE TO COMPARE REGISTERS
11445                                   ;*GOOD DATA IS ALREADY SAVED IN 'SAVERE'
11446                                   ;*TEST DATA IS IN THE REGISTERS
11447                                   ;*CALL IS
11448                                   ;*      JSR     R0,@#COMREG
11449                                   ;*      SAVERE                  ;GOOD DATA
11450                                   ;*      RHCS1                   ;ADDRESS OF ADDRESS TEST DATA
11451                                   ;*      N.                      ;RETURN FOR ERROR
11452                                   ;*      RG                      ;RETURN FOR GOOD COMPARISON
11453                                   ;*ON RETURN WITH ERROR '$GDDAT' HAS GOOD DATA, '$BDDAT' HAS BAD DATA
11454                                   ;*'REGADR' HAS REGISTER ADDRESS
11455
11456  042512                           COMREG:
11457  042512  010146                          MOV     R1,-(SP)        ;;PUSH R1 ON STACK
11458  042514  010246                          MOV     R2,-(SP)        ;;PUSH R2 ON STACK
11459  042516  010346                          MOV     R3,-(SP)        ;;PUSH R3 ON STACK
11460  042520  010446                          MOV     R4,-(SP)        ;;PUSH R4 ON STACK
11461  042522  010546                          MOV     R5,-(SP)        ;;PUSH R5 ON STACK
11462  042524  012001                          MOV     (R0)+,R1        ;R1 HAS ADDRESS OF GOOD DATA
11463  042526  012002                          MOV     (R0)+,R2        ;R2 HAS ADDRESS OF ADDRESS OF TEST DATA
11464  042530  012003                          MOV     (R0)+,R3        ;R3 HAS NUMBER OF WORDS
11465  042532  012004                          MOV     (R0)+,R4        ;R4 HAS RETURN FOR ERROR
11466  042534  011000                          MOV     (R0),R0         ;R0 HAS RETURN ON NO ERROR
11467                                   ;*NOW SAVE REGISTERS
11468  042536  004737  043436                  JSR     PC,@#PUTREG     ;SAVE REGISTERS
11469  042542  113737  004637  002401          MOVB    @#SAVERE+25,@#AS+1;MAKE UPPER BYTE OF RHAS SAME
11470  042550  012705  177776                  MOV     #-2,R5          ;PRESET R5 TO -2
11471                                   ;*NOW COMPARES WILL MADE
11472  042554  062705  000002           1$:     ADD     #2,R5           ;INCREMENT TO INDEX
11473  042560  022122                          CMP     (R1)+,(R2)+     ;COMPARE REGISTER CONTENTS
11474  042562  001420                          BEQ     2$              ;BRANCH IF GOOD
11475  042564  014137  001124                  MOV     -(R1),@#$GDDAT  ;SAVE GOOD DATA
11476  042570  014237  001126                  MOV     -(R2),@#$BDDAT  ;SAVE BAD DATA
11477  042574  016537  002272  004600          MOV     RHWC(R5),@#REGADR       ;SAVE ADDRESS OF FAILING REGISTER
11478  042602  004714                          JSR     PC,@R4          ;RETURN TO MAIN PROGRAM
11479                                                                  ;TO PRINT ERROR
11480  042604  022122                          CMP     (R1)+,(R2)+     ;UNDO -(R1) AND -(R2) FOR ERRORS
11481  042606  017746  136326                  MOV     @SWR,-(SP)      ;GET SWITCH SETTING
11482  042612  042716  177177                  BIC     #^C600,(SP)     ;KEEP ONLY SWITCH 7 AND 8
11483  042616  022726  000200                  CMP     #SW07,(SP)+     ;IS 7 SET AND 8 DOWN
11484  042622  001402                          BEQ     3$              ;BRANCH OUT IF YES
11485  042624  005303                   2$:    DEC     R3              ;ARE ALL COMPARES DONE
11486  042626  001352                          BNE     1$              ;BRANCH IF NOT COMPLETE
11487
11488  042630                           3$:
11489  042630  012605                          MOV     (SP)+,R5        ;;POP STACK INTO R5
11490  042632  012604                          MOV     (SP)+,R4        ;;POP STACK INTO R4
11491  042634  012603                          MOV     (SP)+,R3        ;;POP STACK INTO R3
11492  042636  012602                          MOV     (SP)+,R2        ;;POP STACK INTO R2
11493  042640  012601                          MOV     (SP)+,R1        ;;POP STACK INTO R1
11494  042642  000200                          RTS     R0              ;RETURN TO MAIN PROGRAM
11495  042644  000000                   4$:    .WORD   0               ;TEMP STORAGE
```

```
11496
11497
11498
11499                                   ;*HERE IS A DETAILED EXPLAINATION OF HOW THE LOOP ON ERROR WORKS.
11500                                   ;*ON HITTING AN ERROR IF THE LOOP ON ERROR SWITCH IS SET, THE
11501                                   ;*PROGRAM GOES BACK - USUALLY BACK TO THE BEGINNING OF THE TEST.
11502
11503                                   ;*WHEN THIS OPERATOR SELECTABLE SCOPE LOOP IS USED THEN THE POINT
11504                                   ;*THE PROGRAM GOES BACK TO CAN BE CHANGED.
11505                                   ;*THE RESTRICTIONS TO THE POINT WHERE THE PROGRAM CAN GO ARE: -
11506                                   ;*1.  IT MUST BE WITHIN THE TEST UNDER CONSIDERATION
11507                                   ;*2.  LOOP ON ERROR SWITCH MUST BE SET
11508                                   ;*3.  THE ERROR MUST OCCUR WITHIN THE TEST UNDER CONSIDERATION
11509                                   ;*IF THE ERROR DOES NOT OCCUR WITHIN THE TEST UNDER CONSIDERATION
11510                                   ;*THE PROGRAM WILL REVERT TO NORMAL OPERATION.  HOWEVER, IF LOOP ON
11511                                   ;*TEST SWITCH IS SET AND THIS OPERATOR SELECTABLE SCOPE LOOP IS USED
11512                                   ;*THEN THE PROGRAM WILL LOOP BACK TO THE SELECTED POINT WHEN IT
11513                                   ;*COMES TO THE END OF THE TEST UNDER CONSIDERATION.
11514                                   ;*
11515                                   ;*AFTER LOOPING FOR SOME TIME IF THE LOOP SWITCH IS PUT DOWN THEN
11516                                   ;*NORMAL OPERATION WILL CONTINUE.
11517
11518  042646  000000                   TESTAD: 0                       ;FIRST ADDRESS OF TEST
11519  042650                           OPERSEL:
11520  042650  005037  177776                   CLR     PS              ;MAKE PROCESSOR STATUS ZERO
11521  042654  012737  177777  047342           MOV     #-1,@#PRITEM    ;CLEAR PREVIOUS ITEM NUMBER
11522  042662  104401  042670                   TYPE    ,65$            ;;TYPE ASCIZ STRING
11523  042666  000421                           BR      64$             ;;GET OVER THE ASCIZ
11524                                   ;;65$:  .ASCIZ  <15><12>/THE PROGRAM WAS IN TEST NUMBER /
11525  042732                           64$:
11526  042732  013746  004604                   MOV     @#TSTNM,-(SP)   ;GET READY TO TYPE TEST
11527  042736  104402                           TYPOC                   ;NUMBER
11528  042740  104401  042746                   TYPE    ,67$            ;;TYPE ASCIZ STRING
11529  042744  000414                           BR      66$             ;;GET OVER THE ASCIZ
11530                                   ;;67$:  .ASCIZ  <15><12>/THE LOOP BACK PC WAS /
11531  042776                           66$:
11532  042776  013746  001110                   MOV     @#$LPERR,-(SP)  ;GET READY TO TYPE LOOP BACK PC
11533  043002  104402                           TYPOC
11534  043004  104401  001223                   TYPE    ,$CRLF
11535  043010  104401  043016                   TYPE    ,69$            ;;TYPE ASCIZ STRING
11536  043014  000430                           BR      68$             ;;GET OVER THE ASCIZ
11537                                   ;;69$:  .ASCIZ  <15><12>/SET SWITCH FOR LOOP ON ERROR OR LOOP ON TEST/
11538  043076                           68$:
11539  043076  104401  043104                   TYPE    ,71$            ;;TYPE ASCIZ STRING
11540  043102  000430                           BR      70$             ;;GET OVER THE ASCIZ
11541                                   ;;71$:  .ASCIZ  <15><12>/TYPE THE FIRST PC OF THE TEST TO BE LOOPED ON/
11542  043164                           70$:
11543  043164  104401  043172                   TYPE    ,73$            ;;TYPE ASCIZ STRING
11544  043170  000422                           BR      72$             ;;GET OVER THE ASCIZ
11545                                   ;;73$:  .ASCIZ  <15><12>/ FOLLOWED BY A CARRIAGE RETURN /<15><12>
11546  043236                           72$:
11547  043236  104412                           RDOCT
11548  043240  062716  000002                   ADD     #2,(SP)         ;GET LPADR
11549  043244  012637  001106                   MOV     (SP)+,@#$LPADR
11550  043250  104401  043256                   TYPE    ,75$            ;;TYPE ASCIZ STRING
11551  043254  000417                           BR      74$             ;;GET OVER THE ASCIZ
```

```
11552                                   ;;75$:  .ASCIZ  <15><12>/TYPE THE PC WHERE YOU WANT/
11553  043314                           74$:
11554  043314  104401  043322                  TYPE    ,77$             ;;TYPE ASCIZ STRING
11555  043320  000440                          BR      76$              ;;GET OVER THE ASCIZ
11556                                   ;;77$:  .ASCIZ  <15><12>/ THE PROGRAM TO LOOP BACK TO FOLLOWED BY A CARRIAGE RETURN /<15
11557  043422                           76$:
11558  043422  104412                          RDOCT
11559  043424  012637  001110                  MOV     (SP)+,@#$LPERR  ;GET LPERR
11560  043430  013746  001106                  MOV     @#$LPADR,-(SP)
11561  043434  000002                          RTI
11562
11563
```

```
11564
11565
11566                                   ;*THIS SAVES THE CONTENTS OF ALL HARDWARE REGISTERS
11567                                   ;*IN MEMORY LOCATIONS TAGED FROM "WC" TO "EC2"
11568
11569                                   ;*THIS IS DONE SO THAT COMPARES ARE DONE WITH SAVED LOCATIONS
11570                                   ;*AND NOT THE REGISTERS THEMSELVES.   THIS WILL MAKE
11571                                   ;*ERROR PRINTOUTS FOR GOOD AND BAD DATA ALWAYS DIFFRENT
11572
11573
11574
11575
11576  043436                   PUTREG:
11577  043436  010046                   MOV     R0,-(SP)          ;;PUSH R0 ON STACK
11578  043440  010146                   MOV     R1,-(SP)          ;;PUSH R1 ON STACK
11579  043442  010246                   MOV     R2,-(SP)          ;;PUSH R2 ON STACK
11580  043444  012700  002272           MOV     #RHWC,R0          ;STARTING ADDRESS OF REG
11581  043450  012701  002354           MOV     #WC,R1            ;STARTING ADDRESS OF WERE SAVED
11582  043454  012702  000022           MOV     #RHCC-RHWC+2/2,R2 ;NUMBER OF REG. INTO R2
11583  043460  013021             10$:  MOV     @(R0)+,(R1)+      ;SAVE HARDWARE REG.
11584  043462  005302                   DEC     R2
11585  043464  001375                   BNE     10$
11586  043466  012602                   MOV     (SP)+,R2          ;;POP STACK INTO R2
11587  043470  012601                   MOV     (SP)+,R1          ;;POP STACK INTO R1
11588  043472  012600                   MOV     (SP)+,R0          ;;POP STACK INTO R0
11589  043474  000207                   RTS     PC
11590
11591
11592
11593                                   ;*THIS IS A DATA  COMMAND SETUP SUBROUTINE
11594                                   ;*THE CALL IS
11595                                   ;*      JSR     R0,@#RUN
11596                                   ;*      C                         ;CYLINDER
11597                                   ;*.BYTE S                         ;SECTOR
11598                                   ;*.BYTE T                         ;TRACK
11599                                   ;*      -W                        ;WORD COUNT
11600                                   ;*      B                         ;BUS ADDRESS
11601                                   ;*      BAI                       ;BUS ADDRESS INHIBIT
11602                                   ;*      FMT22!ECI!HCI             ;FMT22=1 =16 BIT WORDS
11603                                   ;*                                ;ECI = ECC CORRECTION INHIBIT
11604                                   ;*                                ;HCI = HEADER COMPARE INHIBIT
11605                                   ;*      COM                       ;COMMAND ADDRESS
11606  043476  012077  136610   RUN:    MOV     (R0)+,@RHCA       ;CYLINDER
11607  043502  012077  136576           MOV     (R0)+,@RHDST      ;DESIRED SECTOR/TRACK
11608  043506  012077  136560           MOV     (R0)+,@RHWC       ;WORD COUNT
11609  043512  012077  136556           MOV     (R0)+,@RHBA       ;BUS ADDRESS
11610  043516  013746  004716           MOV     @#UNIT,-(SP)      ;GET UNIT NO
11611  043522  052016                   BIS     (R0)+,(SP)        ;SET BUS ADDRESS INHIBIT
11612  043524  012677  136546           MOV     (SP)+,@RHCS2      ;UNIT NO AND BAI TO RHCS2
11613  043530  012077  136554           MOV     (R0)+,@RHOF       ;FORMAT, ECC INHIBIT, HEADER
11614                                                             ;COMPARE, IF THERE
11615  043534  013077  136540           MOV     @(R0)+,@RHCS1     ;COMMAND IN RHCS1
11616  043540  000200                   RTS     R0                ;RETURN TO MAIN PROGRAM
11617
11618
```

```
11619
11620
11621                                   ;*THIS IS A SUBROUTINE TO COMPARE TWO BLOCKS IN MEMORY
11622
11623                                   ;*R1 HAS GOOD DATA BUFFER ADDRESS
11624                                   ;*R2 HAS TEST DATA BUFFER ADDRESS
11625                                   ;*R5 HAS ADDRESS OF RETURN ON ERROR
11626                                   ;*R3 HAS NUMBER OF WORDS TO BE COMPARED
11627                                   ;*R4 HAS ONE MORE THAN NUMBER OF WORDS TO BE COMPARED
11628
11629                                   ;*CALL IS:
11630                                   ;*      JSR     R0,@#COMPAR
11631                                   ;*      G                       ;ADDRESS OF GOOD DATA
11632                                   ;*      T                       ;ADDRESS OF TEST DATA
11633                                   ;*      N                       ;NUMBER OF WORDS TO BE COMPARED
11634                                   ;*      RE                      ;RETURN ON ERROR
11635                                   ;*      RG                      ;RETURN ON NO ERROR
11636
11637
11638
11639  043542                           COMPAR:
11640  043542  010146                           MOV     R1,-(SP)        ;;PUSH R1 ON STACK
11641  043544  010246                           MOV     R2,-(SP)        ;;PUSH R2 ON STACK
11642  043546  010346                           MOV     R3,-(SP)        ;;PUSH R3 ON STACK
11643  043550  010446                           MOV     R4,-(SP)        ;;PUSH R4 ON STACK
11644  043552  010546                           MOV     R5,-(SP)        ;;PUSH R5 ON STACK
11645  043554  012001                           MOV     (R0)+,R1        ;ADDRESS OF GOOD DATA BUFFER
11646  043556  012002                           MOV     (R0)+,R2        ;ADDRESS OF TEST DATA BUFFER
11647  043560  012003                           MOV     (R0)+,R3        ;NO OF WORDS TO BE COMPARED
11648  043562  012005                           MOV     (R0)+,R5        ;RETURN ON ERROR
11649  043564  011000                           MOV     (R0),R0         ;RETURN ON NO ERROR
11650  043566  010304                           MOV     R3,R4           ;NO OF WORDS TO BE COMPARED
11651  043570  005204                           INC     R4
11652  043572  010437  004602           1$:     MOV     R4,@#ERWORD     ;FOR ERROR WORD NO
11653  043576  022122                           CMP     (R1)+,(R2)+     ;COMPARE GOOD WITH TEST DATA
11654  043600  001417                           BEQ     2$              ;BRANCH IF GOOD
11655
11656  043602  014137  001124                   MOV     -(R1),@#$GDDAT  ;GOOD DATA
11657  043606  014237  001126                   MOV     -(R2),@#$BDDAT  ;BAD DATA
11658  043612  160337  004602                   SUB     R3,@#ERWORD     ;ERROR WORD NO.
11659  043616  004715                           JSR     PC,@R5          ;RETURN TO PRINT ERROR
11660  043620  022122                           CMP     (R1)+,(R2)+     ;UNDO -(R1) AND -(R2) FOR ERRORS
11661  043622  017746  135312                   MOV     @SWR,-(SP)      ;GET SWITCH SETTING
11662  043626  042716  177177                   BIC     #^C600,(SP)     ;KEEP ONLY SWITCH 7 AND 8
11663  043632  022726  000200                   CMP     #SW07,(SP)+     ;IS 7 SET AND 8 RESET
11664  043636  001402                           BEQ     3$              ;BRANCH OUT IF YES
11665  043640  005303                   2$:     DEC     R3              ;COUNT
11666  043642  001353                           BNE     1$              ;BRANCH IF ALL NOT DEVICE
11667  043644                           3$:
11668  043644  012605                           MOV     (SP)+,R5        ;;POP STACK INTO R5
11669  043646  012604                           MOV     (SP)+,R4        ;;POP STACK INTO R4
11670  043650  012603                           MOV     (SP)+,R3        ;;POP STACK INTO R3
11671  043652  012602                           MOV     (SP)+,R2        ;;POP STACK INTO R2
11672  043654  012601                           MOV     (SP)+,R1        ;;POP STACK INTO R1
11673  043656  000200                           RTS     R0              ;RETURN TO MAIN PROGRAM
```

```
11674
11675
11676                                   ;*      THIS ROUTINE WILL ALLOW THE CHANGE OF THE BASE
11677                                   ;*      ADDRESS FROM 176700 TO ANY TYPED VALUE
11678
11679  043660                           BASECH:
11680  043660  104401  043666                   TYPE    ,65$            ;;TYPE ASCIZ STRING
11681  043664  000425                           BR      64$             ;;GET OVER THE ASCIZ
11682                                   ;;65$:  .ASCIZ  <15><12>/PRESENT BASE ADDRESS OF REGISTERS IS   /
11683  043740                           64$:
11684  043740  013746  002300                   MOV     @#RHCS1,-(SP)   ;GET READY TO TYPE OLD BASE
11685  043744  104402                           TYPOC
11686  043746  104401  043754                   TYPE    ,67$            ;;TYPE ASCIZ STRING
11687  043752  000425                           BR      66$             ;;GET OVER THE ASCIZ
11688                                   ;;67$:  .ASCIZ  <15><12>/TYPE NEW BASE ADDRESS FOLLOWED BY 'CR' /
11689  044026                           66$:
11690  044026  004737  045734                   JSR     PC,@#$TKINT     ;INITIALIZE THE TTY KEYBOARD
11691  044032  104412                           RDOCT
11692  044034  012700  002270                   MOV     #RHDB,R0        ;GET STARTING ADDRESS OF REGISTERS
11693  044040  012701  000026                   MOV     #22.,R1         ;NUMBER OF REGISTERS
11694  044044  012737  044644  000004           MOV     #ADTIMO,@#4     ;SET TRAP CATCHER TO CHECK THIS ADDRESS
11695  044052  021637  002300                   CMP     @SP,@#RHCS1     ;NEW ADDRESS?
11696  044056  001407                           BEQ     1$              ;NO, OLD ONE JUST RETYPED.
11697  044060  005776  000000                   TST     @0(SP)          ;OK, SO ACCESS THIS NEW ADDRESS
11698  044064  163716  002300                   SUB     @#RHCS1,@SP     ;GET THE ADDRESS OFFSET
11699  044070  061620                   2$:     ADD     @SP,(R0)+       ;AND PLUG IT IN.
11700  044072  005301                           DEC     R1              ;ONE LESS REGISTER TO GO
11701  044074  001375                           BNE     2$              ;BUT WE'RE NOT DONE YET.
11702  044076                           1$:
11703  044076  104401  044104                   TYPE    ,69$            ;;TYPE ASCIZ STRING
11704  044102  000417                           BR      68$             ;;GET OVER THE ASCIZ
11705                                   ;;69$:  .ASCIZ  <15><12>/PRESENT VECTOR ADDRESS IS /
11706  044142                           68$:
11707  044142  013746  002266                   MOV     @#RPVEC,-(SP)   ;GET READY TO TYPE OLD VECTOR ADDRESS
11708  044146  104402                           TYPOC
11709  044150  104401  044156                   TYPE    ,71$            ;;TYPE ASCIZ STRING
11710  044154  000437                           BR      70$             ;;GET OVER THE ASCIZ
11711                                   ;;71$:  .ASCIZ  <15><12>/TYPE NEW VECTOR ADDRESS OR RETYPE OLD ONE FOLLOWED BY "CR" /
11712  044254                           70$:
11713  044254  104412                           RDOCT
11714  044256  012637  002266                   MOV     (SP)+,@#RPVEC   ;SETUP VECTOR ADDRESS
11715  044262  104401  044270                   TYPE    ,73$            ;;TYPE ASCIZ STRING
11716  044266  000416                           BR      72$             ;;GET OVER THE ASCIZ
11717                                   ;;73$:  .ASCIZ  <15><12>/NEW BASE WILL REMAIN   - /
11718  044324                           72$:
11719  044324  013746  002300                   MOV     @#RHCS1,-(SP)
11720  044330  104402                           TYPOC
11721  044332  104401  044340                   TYPE    ,75$            ;;TYPE ASCIZ STRING
11722  044336  000416                           BR      74$             ;;GET OVER THE ASCIZ
11723                                   ;;75$:  .ASCIZ  <15><12>/NEW VECTOR WILL REMAIN - /
11724  044374                           74$:
11725  044374  013746  002266                   MOV     @#RPVEC,-(SP)
11726  044400  104402                           TYPOC
11727  044402  104401  044410                   TYPE    ,77$            ;;TYPE ASCIZ STRING
11728  044406  000417                           BR      76$             ;;GET OVER THE ASCIZ
11729                                   ;;77$:  .ASCIZ  <15><12>/UNTIL PROGRAM IS RELOADED./
```

```
11730  044446
11731  044446  104401  044454          76$:
11732  044452  000402                          TYPE    ,79$            ;;TYPE ASCIZ STRING
11733                                          BR      78$             ;;GET OVER THE ASCIZ
11734  044460                          ;;79$:  .ASCIZ  <15><12>/ /
11735  044460  104401  044466          78$:
11736  044464  000424                          TYPE    ,81$            ;;TYPE ASCIZ STRING
11737                                          BR      80$             ;;GET OVER THE ASCIZ
11738  044536                          ;;81$:  .ASCIZ  <15><12>/UNLESS HALTED AND MANUALLY RESTARTED,/
11739  044536  104401  044544          80$:
11740  044542  000426                          TYPE    ,83$            ;;TYPE ASCIZ STRING
11741                                          BR      82$             ;;GET OVER THE ASCIZ
11742  044620                          ;;83$:  .ASCIZ  <15><12>/PROGRAM WILL AUTOMATICALLY RESTART FROM /
11743  044620  012746  000200          82$:
11744  044624  104402                          MOV     #RA,-(SP)
11745  044626  104401  044634                  TYPOC
11746  044632  000402                          TYPE    ,85$            ;;TYPE ASCIZ STRING
11747                                          BR      84$             ;;GET OVER THE ASCIZ
11748  044640                          ;;85$:  .ASCIZ  <15><12>/ /
11749  044640  000137  005012          84$:
11750  044644                                  JMP     @#BEGIN         ;RESTART, TO RUN ALL DRIVES
11751  044644  104401  044652          ADTIMO:
11752  044650  000426                          TYPE    ,65$            ;;TYPE ASCIZ STRING
11753                                          BR      64$             ;;GET OVER THE ASCIZ
11754  044726                          ;;65$:  .ASCIZ  <15><12><377>/THE SELECTED ADDRESS DID NOT RESPOND!  /
11755  044726  022626                  64$:
11756  044730  000137  043660                  CMP     (SP)+,(SP)+     ;RESTORE THE STACK
11757                                          JMP     @#BASECH        ;AND DO THE WHOLE THING AGAIN!
11758
```

```
11759                                   ;;*****************************************************************
11760
11761  044734                           RPVECT:
11762  044734  104401  044742                  TYPE    ,65$            ;;TYPE ASCIZ STRING
11763  044740  000411                          BR      64$             ;;GET OVER THE ASCIZ
11764                                   ;;65$:  .ASCIZ  /TRAPED FROM PC = /
11765  044764                           64$:
11766  044764  104402                          TYPOC                           ;TYPE FROM PC
11767  044766  012777  044734  135272          MOV     #RPVECT,@RPVEC          ;RESTORE TRAP RP04 VECTOR
11768  044774  000000                          HALT                            ;CHANGE TO CONTINUE
11769                                   ;;*****************************************************************
11770
11771
11772
11773
```

```
                                .SBTTL  SCOPE HANDLER ROUTINE

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

 044776                         $SCOPE:
 044776  104407                        CKSWR                   ;;TEST  FOR CHANGE IN SOFT-SWR
 045000  032777  040000  134132 1$:     BIT     #BIT14,@SWR     ;;LOOP ON PRESENT TEST?
 045006  001111                        BNE     $OVER           ;;YES IF SW14=1
                                ;#####START OF CODE FOR THE XOR TESTER#####
 045010  000416                 $XTSTR: BR      6$              ;;IF RUNNING ON THE "XOR" TESTER CHANGE
                                                                ;;THIS INSTRUCTION TO A "NOP" (NOP=240)
 045012  013746  000004                 MOV     @#ERRVEC,-(SP)  ;;SAVE THE CONTENTS OF THE ERROR VECTOR
 045016  012737  045036  000004         MOV     #5$,@#ERRVEC    ;;SET FOR TIMEOUT
 045024  005737  177060                 TST     @#177060        ;;TIME OUT ON XOR?
 045030  012637  000004                 MOV     (SP)+,@#ERRVEC  ;;RESTORE THE ERROR VECTOR
 045034  000463                        BR      $SVLAD          ;;GO TO THE NEXT TEST
 045036  022626                 5$:     CMP     (SP)+,(SP)+     ;;CLEAR THE STACK AFTER A TIME OUT
 045040  012637  000004                 MOV     (SP)+,@#ERRVEC  ;;RESTORE THE ERROR VECTOR
 045044  000423                        BR      7$              ;;LOOP ON THE PRESENT TEST
 045046                         6$:;#####END OF CODE FOR THE XOR TESTER#####
 045046  032777  000400  134064         BIT     #BIT08,@SWR     ;;LOOP ON SPEC. TEST?
 045054  001404                        BEQ     2$              ;;BR IF NO
 045056  127737  134056  001102         CMPB    @SWR,$TSTNM     ;;ON THE RIGHT TEST?    SWR<7:0>
 045064  001462                        BEQ     $OVER           ;;BR IF YES
 045066  105737  001103         2$:    TSTB    $ERFLG          ;;HAS AN ERROR OCCURRED?
 045072  001421                        BEQ     3$              ;;BR IF NO
 045074  123737  001115  001103         CMPB    $ERMAX,$ERFLG   ;;MAX. ERRORS FOR THIS TEST OCCURRED?
 045102  101015                        BHI     3$              ;;BR IF NO
 045104  032777  001000  134026         BIT     #BIT09,@SWR     ;;LOOP ON ERROR?
 045112  001404                        BEQ     4$              ;;BR IF NO
 045114  013737  001110  001106 7$:     MOV     $LPERR,$LPADR   ;;SET LOOP ADDRESS TO LAST SCOPE
 045122  000443                        BR      $OVER
 045124  105037  001103         4$:    CLRB    $ERFLG          ;;ZERO THE ERROR FLAG
 045130  005037  001212                 CLR     $TIMES          ;;CLEAR THE NUMBER OF ITERATIONS TO MAKE
 045134  000415                        BR      1$              ;;ESCAPE TO THE NEXT TEST
 045136  032777  004000  133774 3$:     BIT     #BIT11,@SWR     ;;INHIBIT ITERATIONS?
 045144  001011                        BNE     1$              ;;BR IF YES
 045146  005737  001100                 TST     $PASS           ;;IF FIRST PASS OF PROGRAM
 045152  001406                        BEQ     1$              ;;      INHIBIT ITERATIONS
 045154  005237  001104                 INC     $ICNT           ;;INCREMENT ITERATION COUNT
 045160  023737  001212  001104         CMP     $TIMES,$ICNT    ;;CHECK THE NUMBER OF ITERATIONS MADE
 045166  002021                        BGE     $OVER           ;;BR IF MORE ITERATION REQUIRED
 045170  012737  000001  001104 1$:     MOV     #1,$ICNT        ;;REINITIALIZE THE ITERATION COUNTER
 045176  013737  045246  001212         MOV     $MXCNT,$TIMES   ;;SET NUMBER OF ITERATIONS TO DO
 045204  105237  001102         $SVLAD: INCB    $TSTNM          ;;COUNT TEST NUMBERS
 045210  011637  001106                 MOV     (SP),$LPADR     ;;SAVE SCOPE LOOP ADDRESS
```

```
 11830  045214  011637  001110                  MOV     (SP),$LPERR     ;;SAVE ERROR LOOP ADDRESS
 11831  045220  005037  001214                  CLR     $ESCAPE         ;;CLEAR THE ESCAPE FROM ERROR ADDRESS
 11832  045224  112737  000001  001115          MOVB    #1,$ERMAX       ;;ONLY ALLOW ONE(1) ERROR ON NEXT TEST
 11833  045232  013777  001102  133702  $OVER:  MOV     $TSTNM,@DISPLAY ;;DISPLAY TEST NUMBER
 11834  045240  013716  001106                  MOV     $LPADR,(SP)     ;;FUDGE RETURN ADDRESS
 11835  045244  000002                          RTI                     ;;FIXES PS
 11836  045246  000004          $MXCNT: 4                               ;;MAX. NUMBER OF ITERATIONS
```

```
                                        .SBTTL  CONVERT BINARY TO DECIMAL AND TYPE ROUTINE

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

045250                                  $TYPDS:
045250  010046                                  MOV     R0,-(SP)        ;;PUSH R0 ON STACK
045252  010146                                  MOV     R1,-(SP)        ;;PUSH R1 ON STACK
045254  010246                                  MOV     R2,-(SP)        ;;PUSH R2 ON STACK
045256  010346                                  MOV     R3,-(SP)        ;;PUSH R3 ON STACK
045260  010546                                  MOV     R5,-(SP)        ;;PUSH R5 ON STACK
045262  012746  020200                         MOV     #20200,-(SP)    ;;SET BLANK SWITCH AND SIGN
045266  016605  000020                         MOV     20(SP),R5       ;;GET THE INPUT NUMBER
045272  100004                                  BPL     1$              ;;BR IF INPUT IS POS.
045274  005405                                  NEG     R5              ;;MAKE THE BINARY NUMBER POS.
045276  112766  000055  000001                 MOVB    #'-,1(SP)       ;;MAKE THE ASCII NUMBER NEG.
045304  005000                          1$:     CLR     R0              ;;ZERO THE CONSTANTS INDEX
045306  012703  045464                         MOV     #$DBLK,R3       ;;SETUP THE OUTPUT POINTER
045312  112723  000040                         MOVB    #' ,(R3)+       ;;SET THE FIRST CHARACTER TO A BLANK
045316  005002                          2$:     CLR     R2              ;;CLEAR THE BCD NUMBER
045320  016001  045454                         MOV     $DTBL(R0),R1    ;;GET THE CONSTANT
045324  160105                          3$:     SUB     R1,R5           ;;FORM THIS BCD DIGIT
045326  002402                                  BLT     4$              ;;BR IF DONE
045330  005202                                  INC     R2              ;;INCREASE THE BCD DIGIT BY 1
045332  000774                                  BR      3$
045334  060105                          4$:     ADD     R1,R5           ;;ADD BACK THE CONSTANT
045336  005702                                  TST     R2              ;;CHECK IF BCD DIGIT=0
045340  001002                                  BNE     5$              ;;FALL THROUGH IF 0
045342  105716                                  TSTB    (SP)            ;;STILL DOING LEADING 0'S?
045344  100407                                  BMI     7$              ;;BR IF YES
045346  106316                          5$:     ASLB    (SP)            ;;MSD?
045350  103003                                  BCC.    6$              ;;BR IF NO
045352  116663  000001  177777                 MOVB    1(SP),-1(R3)    ;;YES--SET THE SIGN
045360  052702  000060                  6$:     BIS     #'0,R2          ;;MAKE THE BCD DIGIT ASCII
045364  052702  000040                  7$:     BIS     #' ,R2          ;;MAKE IT A SPACE IF NOT ALREADY A DIGIT
045370  110223                                  MOVB    R2,(R3)+        ;;PUT THIS CHARACTER IN THE OUTPUT BUFFER
045372  005720                                  TST     (R0)+           ;;JUST INCREMENTING
045374  020027  000010                         CMP     R0,#10          ;;CHECK THE TABLE INDEX
045400  002746                                  BLT     2$              ;;GO DO THE NEXT DIGIT
045402  003002                                  BGT     8$              ;;GO TO EXIT
045404  010502                                  MOV     R5,R2           ;;GET THE LSD
045406  000764                                  BR      6$              ;;GO CHANGE TO ASCII
045410  105726                          8$:     TSTB    (SP)+           ;;WAS THE LSD THE FIRST NON-ZERO?
045412  100003                                  BPL     9$              ;;BR IF NO
045414  116663  177777  177776                 MOVB    -1(SP),-2(R3)   ;;YES--SET THE SIGN FOR TYPING
045422  105013                          9$:     CLRB    (R3)            ;;SET THE TERMINATOR
045424  012605                                  MOV     (SP)+,R5        ;;POP STACK INTO R5
045426  012603                                  MOV     (SP)+,R3        ;;POP STACK INTO R3
045430  012602                                  MOV     (SP)+,R2        ;;POP STACK INTO R2
```

```
11893  045432  012601                          MOV     (SP)+,R1        ;;POP STACK INTO R1
11894  045434  012600                          MOV     (SP)+,R0        ;;POP STACK INTO R0
11895  045436  104401  045464                  TYPE    ,$DBLK          ;;NOW TYPE THE NUMBER
11896  045442  016666  000002  000004          MOV     2(SP),4(SP)     ;;ADJUST THE STACK
11897  045450  012616                          MOV     (SP)+,(SP)
11898  045452  000002                          RTI                     ;;RETURN TO USER
11899  045454  023420                $DTBL:    10000.
11900  045456  001750                          1000.
11901  045460  000144                          100.
11902  045462  000012                          10.
11903  045464  000004                $DBLK:    .BLKW   4
```

```
11904                                   .SBTTL  TYPE ROUTINE
11905
11906                                   ;;*********************************************************************
11907                                   ;*ROUTINE TO TYPE ASCIZ MESSAGE. MESSAGE MUST TERMINATE WITH A 0 BYTE.
11908                                   ;*THE ROUTINE WILL INSERT A NUMBER OF NULL CHARACTERS AFTER A LINE FEED.
11909                                   ;*NOTE1:        $NULL CONTAINS THE CHARACTER TO BE USED AS THE FILLER CHARACTER.
11910                                   ;*NOTE2:        $FILLS CONTAINS THE NUMBER OF FILLER CHARACTERS REQUIRED.
11911                                   ;*NOTE3:        $FILLC CONTAINS THE CHARACTER TO FILL AFTER.
11912                                   ;*
11913                                   ;*CALL:
11914                                   ;*1) USING A TRAP INSTRUCTION
11915                                   ;*      TYPE    ,MESADR         ;;MESADR IS FIRST ADDRESS OF AN ASCIZ STRING
11916                                   ;*OR
11917                                   ;*      TYPE
11918                                   ;*      MESADR
11919                                   ;*
11920
11921  045474  105737  001157          $TYPE:  TSTB    $TPFLG          ;;IS THERE A TERMINAL?
11922  045500  100002                          BPL     1$              ;;BR IF YES
11923  045502  000000                          HALT                    ;;HALT HERE IF NO TERMINAL
11924  045504  000407                          BR      3$              ;;LEAVE
11925  045506  010046                  1$:     MOV     R0,-(SP)        ;;SAVE R0
11926  045510  017600  000002                  MOV     @2(SP),R0       ;;GET ADDRESS OF ASCIZ STRING
11927  045514  112046                  2$:     MOVB    (R0)+,-(SP)     ;;PUSH CHARACTER TO BE TYPED ONTO STACK
11928  045516  001005                          BNE     4$              ;;BR IF IT ISN'T THE TERMINATOR
11929  045520  005726                          TST     (SP)+           ;;IF TERMINATOR POP IT OFF THE STACK
11930  045522  012600                  60$:    MOV     (SP)+,R0        ;;RESTORE R0
11931  045524  062716  000002          3$:     ADD     #2,(SP)         ;;ADJUST RETURN PC
11932  045530  000002                          RTI                     ;;RETURN
11933  045532  122716  000011          4$:     CMPB    #HT,(SP)        ;;BRANCH IF <HT>
11934  045536  001430                          BEQ     8$
11935  045540  122716  000200                  CMPB    #CRLF,(SP)      ;;BRANCH IF NOT <CRLF>
11936  045544  001006                          BNE     5$
11937  045546  005726                          TST     (SP)+           ;;POP  <CR><LF> EQUIV
11938  045550  104401                          TYPE                    ;;TYPE A CR AND LF
11939  045552  001223                          $CRLF
11940  045554  105037  045710                  CLRB    $CHARCNT        ;;CLEAR CHARACTER COUNT
11941  045560  000755                          BR      2$              ;;GET NEXT CHARACTER
11942  045562  004737  045644          5$:     JSR     PC,$TYPEC       ;;GO TYPE THIS CHARACTER
11943  045566  123726  001156          6$:     CMPB    $FILLC,(SP)+    ;;IS IT TIME FOR FILLER CHARS.?
11944  045572  001350                          BNE     2$              ;;IF NO GO GET NEXT CHAR.
11945  045574  013746  001154                  MOV     $NULL,-(SP)     ;;GET # OF FILLER CHARS. NEEDED
11946                                                                  ;;AND THE NULL CHAR.
11947  045600  105366  000001          7$:     DECB    1(SP)           ;;DOES A NULL NEED TO BE TYPED?
11948  045604  002770                          BLT     6$              ;;BR IF NO--GO POP THE NULL OFF OF STACK
11949  045606  004737  045644                  JSR     PC,$TYPEC       ;;GO TYPE A NULL
11950  045612  105337  045710                  DECB    $CHARCNT        ;;DO NOT COUNT AS A COUNT
11951  045616  000770                          BR      7$              ;;LOOP
11952
11953                                   ;HORIZONTAL TAB PROCESSOR
11954
11955  045620  112716  000040          8$:     MOVB    #' ,(SP)        ;;REPLACE TAB WITH SPACE
11956  045624  004737  045644          9$:     JSR     PC,$TYPEC       ;;TYPE A SPACE
11957  045630  132737  000007  045710          BITB    #7,$CHARCNT     ;;BRANCH IF NOT AT
11958  045636  001372                          BNE     9$              ;;TAB STOP
11959  045640  005726                          TST     (SP)+           ;;POP SPACE OFF STACK
```

```
11960  045642  000724                          BR      2$              ;;GET NEXT CHARACTER
11961  045644  105777  133300          $TYPEC: TSTB    @$TPS           ;;WAIT UNTIL PRINTER IS READY
11962  045650  100375                          BPL     $TYPEC
11963  045652  116677  000002  133272          MOVB    2(SP),@$TPB     ;;LOAD CHAR TO BE TYPED INTO DATA REG.
11964  045660  122766  000015  000002          CMPB    #CR,2(SP)       ;;IS CHARACTER A CARRIAGE RETURN?
11965  045666  001003                          BNE     1$              ;;BRANCH IF NO
11966  045670  105037  045710                  CLRB    $CHARCNT        ;;YES--CLEAR CHARACTER COUNT
11967  045674  000406                          BR      $TYPEX          ;;EXIT
11968  045676  122766  000012  000002  1$:     CMPB    #LF,2(SP)       ;;IS CHARACTER A LINE FEED?
11969  045704  001402                          BEQ     $TYPEX          ;;BRANCH IF YES
11970  045706  105227                          INCB    (PC)+           ;;COUNT THE CHARACTER
11971  045710  000000          $CHARCNT:.WORD  0               ;;CHARACTER COUNT STORAGE
11972  045712  000207          $TYPEX: RTS     PC
11973
```

```
                                        .SBTTL  TTY INPUT ROUTINE
                                        ;;*****************************************************************
                                        .ENABL  LSB
045714  000000                  $TKCNT: .WORD   0               ;;NUMBER OF ITEMS IN QUEUE
045716  000000                  $TKQIN: .WORD   0               ;;INPUT POINTER
045720  000000                  $TKQOUT: .WORD  0               ;;OUTPUT POINTER
045722  000011                  $TKQSRT: .BLKB  9.              ;;TTY KEYBOARD QUEUE
        045733                  $TKQEND=.
        045734                  .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
                                ;
045734  005037  045714          $TKINT: CLR     $TKCNT          ;;CLEAR COUNT OF ITEMS IN QUEUE
045740  012737  045722  045716          MOV     #$TKQSRT,$TKQIN ;;MOVE THE STARTING ADDRESS OF THE
045746  013737  045716  045720          MOV     $TKQIN,$TKQOUT  ;;QUEUE INTO THE INPUT & OUTPUT POINTERS.
045754  012737  046004  000060          MOV     #$TKSRV,@#TKVEC ;;INITIALIZE THE KEYBOARD VECTOR
045762  012737  000200  000062          MOV     #200,@#TKVEC+2  ;;"BR" LEVEL 4
045770  005777  133152                  TST     @$TKB           ;;CLEAR DONE FLAG
045774  012777  000100  133142          MOV     #100,@$TKS      ;;ENABLE TTY KEYBOARD INTERRUPT
046002  000207                  $TKEXT: 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)
                                ;
046004  117746  133136          $TKSRV: MOVB    @$TKB,-(SP)     ;;PICKUP THE CHARACTER
046010  042716  177600                  BIC     #^C177,(SP)     ;;STRIP THE JUNK
046014  021627  000003                  CMP     (SP),#3         ;;IS IT A CONTROL C?
046020  001007                          BNE     1$              ;;BRANCH IF NO
046022  104401  046773                  TYPE    ,$CNTLC         ;;TYPE A CONTROL-C (^C)
046026  004737  045734                  JSR     PC,$TKINT       ;;INIT THE KEYBOARD
046032  005726                          TST     (SP)+           ;;CLEAN UP STACK
046034  000137  042650                  JMP     OPERSEL         ;;CONTROL C RESTART
046040  021627  000007          1$:     CMP     (SP),#7         ;;IS IT A CONTROL G?
046044  001004                          BNE     2$              ;;BRANCH IF NO
046046  022737  000176  001140          CMP     #SWREG,SWR      ;;IS SOFT-SWR SELECTED?
046054  001500                          BEQ     6$              ;;GO TO SWR CHANGE

046056                          2$:
046056  022737  000011  045714          CMP     #9.,$TKCNT      ;;IS THE QUEUE FULL?
046064  001004                          BNE     3$              ;;BRANCH IF NO
046066  104401  001216                  TYPE    ,$BELL          ;;RING THE TTY BELL
046072  005726                          TST     (SP)+           ;;CLEAN CHARACTER OFF OF STACK
046074  000451                          BR      5$              ;;EXIT
046076  021627  000023          3$:     CMP     (SP),#23        ;;IS IT A CONTROL-S?
046102  001021                          BNE     32$             ;;BRANCH IF NO
```

```
12030  046104  005077  133034                CLR     @$TKS           ;;DISABLE TTY KEYBOARD INTERRUPTS
12031  046110  005726                        TST     (SP)+           ;;CLEAN CHAR OFF STACK
12032  046112  105777  133026        31$:    TSTB    @$TKS           ;;WAIT FOR A CHAR
12033  046116  100375                        BPL     31$             ;;LOOP UNTIL ITS THERE
12034  046120  117746  133022                MOVB    @$TKB,-(SP)     ;;GET THE CHARACTER
12035  046124  042716  177600                BIC     #^C177,(SP)     ;;MAKE IT 7-BIT ASCII
12036  046130  022627  000021                CMP     (SP)+,#21       ;;IS IT A CONTROL-Q?
12037  046134  001366                        BNE     31$             ;;BRANCH IF NO
12038  046136  012777  000100  133000        MOV     #100,@$TKS      ;;REENABLE TTY KEYBOARD INTERRUPTS
12039  046144  000002                        RTI                     ;;RETURN
12040  046146  005237  045714        32$:    INC     $TKCNT          ;;COUNT THIS CHARACTER
12041  046152  021627  000140                CMP     (SP),#140       ;;IS IT UPPER CASE?
12042  046156  002405                        BLT     4$              ;;BRANCH IF YES
12043  046160  021627  000175                CMP     (SP),#175       ;;IS IT A SPECIAL CHAR?
12044  046164  003002                        BGT     4$              ;;BRANCH IF YES
12045  046166  042716  000040                BIC     #40,(SP)        ;;MAKE IT UPPER CASE
12046  046172  112677  177520        4$:     MOVB    (SP)+,@$TKQIN   ;;AND PUT IT IN QUEUE
12047  046176  005237  045716                INC     $TKQIN          ;;UPDATE THE POINTER
12048  046202  023727  045716  045733        CMP     $TKQIN,#$TKQEND ;;GO OFF THE END?
12049  046210  001003                        BNE     5$              ;;BRANCH IF NO
12050  046212  012737  045722  045716        MOV     #$TKQSRT,$TKQIN ;;RESET THE POINTER
12051  046220  000002                5$:     RTI                     ;;RETURN
12052
12053                                ;;**********************************************************************
12054                                ;*SOFTWARE SWITCH REGISTER CHANGE ROUTINE.
12055                                ;*ROUTINE IS ENTERED  FROM THE TRAP HANDLER, AND WILL
12056                                ;*SERVICE THE TEST FOR CHANGE IN SOFTWARE SWITCH REGISTER TRAP
12057                                ;*CALL WHEN OPERATING IN TTY INTERRUPT MODE.
12058  046222  022737  000176  001140 $CKSWR: CMP     #SWREG,SWR      ;;IS THE SOFT-SWR SELECTED
12059  046230  001124                        BNE     15$             ;;EXIT IF NOT
12060  046232  105777  132706                TSTB    @$TKS           ;;IS A CHAR WAITING?
12061  046236  100121                        BPL     15$             ;;IF NOT, EXIT
12062  046240  117746  132702                MOVB    @$TKB,-(SP)     ;;YES
12063  046244  042716  177600                BIC     #^C177,(SP)     ;;MAKE IT  7-BIT ASCII
12064  046250  021627  000007                CMP     (SP),#7         ;;IS IT A CONTROL-G?
12065  046254  001300                        BNE     2$              ;;IF NOT, PUT IT IN THE TTY QUEUE
12066                                                                ;;AND EXIT
12067
12068                                ;;**********************************************************************
12069                                ;*CONTROL IS PASSED TO THIS POINT FROM EITHER THE TTY INTERRUPT SERVICE
12070                                ;*ROUTINE OR FROM THE SOFTWARE SWITCH REGISTER TRAP CALL, AS A RESULT OF A
12071                                ;*CONTROL-G BEING TYPED, AND THE SOFTWARE SWITCH REGISTER BEING SELECTED.
12072  046256  123727  001134  000001 6$:     CMPB    $AUTOB,#1       ;;ARE WE RUNNING IN AUTO-MODE?
12073  046264  001674                        BEQ     2$              ;;BRANCH IF YES
12074  046266  005726                        TST     (SP)+           ;;CLEAR CONTROL-G OFF STACK
12075  046270  004737  045734                JSR     PC,$TKINT       ;;FLUSH THE TTY INPUT QUEUE
12076  046274  005077  132644                CLR     @$TKS           ;;DISABLE TTY KEYBOARD INTERRUPTS
12077  046300  112737  000001  001135        MOVB    #1,$INTAG       ;;SET INTERRUPT MODE INDICATOR
12078
12079  046306  104401  047005                TYPE    ,$CNTLG         ;;ECHO THE CONTROL-G (^G)
12080  046312  104401  047012        $GTSWR: TYPE    ,$MSWR          ;;TYPE CURRENT CONTENTS
12081  046316  013746  000176                MOV     SWREG,-(SP)     ;;SAVE SWREG FOR TYPEOUT
12082  046322  104402                        TYPOC                   ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
12083  046324  104401  047023                TYPE    ,$MNEW          ;;PROMPT FOR NEW SWR
12084  046330  005046                19$:    CLR     -(SP)           ;;CLEAR COUNTER
12085  046332  005046                        CLR     -(SP)           ;;THE NEW SWR
```

```
12086  046334  105777  132604          7$:     TSTB    @$TKS           ;;CHAR THERE?
12087  046340  100375                          BPL     7$              ;;IF NOT TRY AGAIN
12088
12089  046342  117746  132600                  MOVB    @$TKB,-(SP)     ;;PICK UP CHAR
12090  046346  042716  177600                  BIC     #^C177,(SP)     ;;MAKE IT 7-BIT ASCII
12091
12092  046352  021627  000003                  CMP     (SP),#3         ;;IS IT A CONTROL-C?
12093  046356  001015                          BNE     9$              ;;BRANCH IF NOT
12094  046360  104401  046773                  TYPE    ,$CNTLC         ;;YES, ECHO CONTROL-C (^C)
12095  046364  062706  000006                  ADD     #6,SP           ;;CLEAN UP STACK
12096  046370  123727  001135  000001          CMPB    $INTAG,#1       ;;REENABLE TTY KEYBOARD INTERRUPTS?
12097  046376  001003                          BNE     8$              ;;BRANCH IF NO
12098  046400  012777  000100  132536          MOV     #100,@$TKS      ;;ALLOW TTY KEYBOARD INTERRUPTS
12099  046406  000137  042650          8$:     JMP     OPERSEL         ;;CONTROL-C RESTART
12100
12101
12102  046412  021627  000025          9$:     CMP     (SP),#25        ;;IS IT A CONTROL-U?
12103  046416  001005                          BNE     10$             ;;BRANCH IF NOT
12104  046420  104401  047000                  TYPE    ,$CNTLU         ;;YES, ECHO CONTROL-U (^U)
12105  046424  062706  000006          20$:    ADD     #6,SP           ;;IGNORE PREVIOUS INPUT
12106  046430  000737                          BR      19$             ;;LET'S TRY IT AGAIN
12107
12108
12109  046432  021627  000015          10$:    CMP     (SP),#15        ;;IS IT A <CR>?
12110  046436  001022                          BNE     16$             ;;BRANCH IF NO
12111  046440  005766  000004                  TST     4(SP)           ;;YES, IS IT THE FIRST CHAR?
12112  046444  001403                          BEQ     11$             ;;BRANCH IF YES
12113  046446  016677  000002  132464          MOV     2(SP),@SWR      ;;SAVE NEW SWR
12114  046454  062706  000006          11$:    ADD     #6,SP           ;;CLEAR UP STACK
12115  046460  104401  001223          14$:    TYPE    ,$CRLF          ;;ECHO <CR> AND <LF>
12116  046464  123727  001135  000001          CMPB    $INTAG,#1       ;;RE-ENABLE TTY KBD INTERRUPTS?
12117  046472  001003                          BNE     15$             ;;BRANCH IF NOT
12118  046474  012777  000100  132442          MOV     #100,@$TKS      ;;RE-ENABLE TTY KBD INTERRUPTS
12119  046502  000002          15$:    RTI                             ;;RETURN
12120  046504  004737  045644          16$:    JSR     PC,$TYPEC       ;;ECHO CHAR
12121  046510  021627  000060                  CMP     (SP),#60        ;;CHAR < 0?
12122  046514  002420                          BLT     18$             ;;BRANCH IF YES
12123  046516  021627  000067                  CMP     (SP),#67        ;;CHAR > 7?
12124  046522  003015                          BGT     18$             ;;BRANCH IF YES
12125  046524  042726  000060                  BIC     #60,(SP)+       ;;STRIP-OFF ASCII
12126  046530  005766  000002                  TST     2(SP)           ;;IS THIS THE FIRST CHAR
12127  046534  001403                          BEQ     17$             ;;BRANCH IF YES
12128  046536  006316                          ASL     (SP)            ;;NO, SHIFT PRESENT
12129  046540  006316                          ASL     (SP)            ;;   CHAR OVER TO MAKE
12130  046542  006316                          ASL     (SP)            ;;   ROOM FOR NEW ONE.
12131  046544  005266  000002          17$:    INC     2(SP)           ;;KEEP COUNT OF CHAR
12132  046550  056616  177776                  BIS     -2(SP),(SP)     ;;SET IN NEW CHAR
12133  046554  000667                          BR      7$              ;;GET THE NEXT ONE
12134  046556  104401  001222          18$:    TYPE    ,$QUES          ;;TYPE ?<CR><LF>
12135  046562  000720                          BR      20$             ;;SIMULATE CONTROL-U
12136                                  .DSABL  LSB
12137
12138
12139                                  ;;**************************************************************************
12140                                  ;*THIS ROUTINE WILL INPUT A SINGLE CHARACTER FROM THE TTY
12141                                  ;*CALL:
```

```
12142                                   ;*      RDCHR                   ;;GET A CHARACTER FROM THE QUEUE
12143                                   ;*      RETURN HERE             ;;CHARACTER IS ON THE STACK
12144                                   ;*                              ;;WITH PARITY BIT STRIPPED OFF
12145                                   ;
12146
12147  046564  011646                  $RDCHR: MOV     (SP),-(SP)      ;;PUSH DOWN THE PC AND
12148  046566  016666  000004  000002          MOV     4(SP),2(SP)     ;;THE PS
12149  046574  005066  000004                  CLR     4(SP)           ;;GET READY FOR A CHARACTER
12150  046600  005046                          CLR     -(SP)           ;;PUT NEW PS ON STACK
12151  046602  012746  046610                  MOV     #64$,-(SP)      ;;PUT NEW PC ON  STACK
12152  046606  000002                          RTI                     ;;POP NEW PC AND PS
12153  046610                          64$:
12154  046610  005737  045714          1$:     TST     $TKCNT          ;;WAIT ON A CHARACTER
12155  046614  001775                          BEQ     1$
12156  046616  005337  045714                  DEC     $TKCNT          ;;DECREMENT THE COUNTER
12157  046622  117766  177072  000004          MOVB    @$TKQOUT,4(SP)  ;;GET ONE CHARACTER
12158  046630  005237  045720                  INC     $TKQOUT         ;;UPDATE THE POINTER
12159  046634  023727  045720  045733          CMP     $TKQOUT,#$TKQEND ;;DID IT GO OFF OF THE END?
12160  046642  001003                          BNE     2$              ;;BRANCH IF NO
12161  046644  012737  045722  045720          MOV     #$TKQSRT,$TKQOUT ;;RESET THE POINTER
12162  046652  000002                  2$:     RTI                     ;;RETURN
12163                                   ;;**********************************************************************
12164                                   ;*THIS ROUTINE WILL INPUT A STRING FROM THE TTY
12165                                   ;*CALL:
12166                                   ;*      RDLIN                   ;;INPUT A STRING FROM THE TTY
12167                                   ;*      RETURN HERE             ;;ADDRESS OF FIRST CHARACTER WILL BE ON THE STACK
12168                                   ;*                              ;;TERMINATOR WILL BE A BYTE OF ALL 0'S
12169
12170  046654  010346                  $RDLIN: MOV     R3,-(SP)        ;;SAVE R3
12171  046656  012703  046762          1$:     MOV     #$TTYIN,R3      ;;GET ADDRESS
12172  046662  022703  046773          2$:     CMP     #$TTYIN+9.,R3   ;;BUFFER FULL?
12173  046666  101405                          BLOS    4$              ;;BR IF YES
12174  046670  104410                          RDCHR                   ;;GO READ ONE CHARACTER FROM THE TTY
12175  046672  112613                          MOVB    (SP)+,(R3)      ;;GET CHARACTER
12176  046674  122713  000177          10$:    CMPB    #177,(R3)       ;;IS IT A RUBOUT
12177  046700  001003                          BNE     3$              ;;SKIP IF NOT
12178  046702  104401  001222          4$:     TYPE    ,$QUES          ;;TYPE A '?'
12179  046706  000763                          BR      1$              ;;CLEAR THE BUFFER AND LOOP
12180  046710  111337  046760          3$:     MOVB    (R3),9$         ;;ECHO THE CHARACTER
12181  046714  104401  046760                  TYPE    ,9$
12182  046720  122723  000015                  CMPB    #15,(R3)+       ;;CHECK FOR RETURN
12183  046724  001356                          BNE     2$              ;;LOOP IF NOT RETURN
12184  046726  105063  177777                  CLRB    -1(R3)          ;;CLEAR RETURN (THE 15)
12185  046732  104401  001224                  TYPE    ,$LF            ;;TYPE A LINE FEED
12186  046736  012603                          MOV     (SP)+,R3        ;;RESTORE R3
12187  046740  011646                          MOV     (SP),-(SP)      ;;ADJUST THE STACK AND PUT ADDRESS OF THE
12188  046742  016666  000004  000002          MOV     4(SP),2(SP)     ;;      FIRST ASCII CHARACTER ON IT
12189  046750  012766  046762  000004          MOV     #$TTYIN,4(SP)
12190  046756  000002                          RTI                     ;;RETURN
12191  046760     000                  9$:     .BYTE   0               ;;STORAGE FOR ASCII CHAR. TO TYPE
12192  046761     000                          .BYTE   0               ;;TERMINATOR
12193  046762  000011                  $TTYIN: .BLKB   9.              ;;RESERVE 9. BYTES FOR TTY INPUT
12194  046773     136  006503  000012  $CNTLC: .ASCIZ  /^C/<15><12>    ;;CONTROL "C"
12195  047000  052536  005015     000  $CNTLU: .ASCIZ  /^U/<15><12>    ;;CONTROL "U"
12196  047005     136  006507  000012  $CNTLG: .ASCIZ  /^G/<15><12>    ;;CONTROL "G"
12197  047012  005015  053523  020122  $MSWR:  .ASCIZ  <15><12>/SWR = /
```

```
 12198  047020  020075     000
 12199  047023     040  047040  053505  $MNEW:  .ASCIZ  /  NEW = /
 12200  047030  036440  000040
 12201                                                    ;FROM THE TTY
```

```
12202                                   .SBTTL  READ AN OCTAL NUMBER FROM THE TTY
12203
12204                                   ;;*****************************************************************
12205                                   ;*THIS ROUTINE WILL READ AN OCTAL (ASCII) NUMBER FROM THE TTY AND
12206                                   ;*CHANGE IT TO BINARY.
12207                                   ;*THE INPUT CHARACTERS WILL BE CHECKED TO INSURED THEY ARE LEGAL
12208                                   ;*OCTAL DIGITS. IF AN ILLEGAL CHARACTER IS READ A "?" WILL BE TYPED
12209                                   ;*FOLLOWED BY A CARRIAGE RETURN-LINE FEED. THE COMPLETE NUMBER MUST
12210                                   ;*THEN BE RETYPED. THE INPUT IS TERMINATED BY TYPING A CARRIAGE RETURN.
12211                                   ;*CALL:
12212                                   ;*      RDOCT                   ;;READ AN OCTAL NUMBER
12213                                   ;*      RETURN HERE             ;;LOW ORDER BITS ARE ON TOP OF THE STACK
12214                                   ;*                              ;;HIGH ORDER BITS ARE IN $HIOCT
12215
12216  047034  011646                  $RDOCT: MOV     (SP),-(SP)      ;;PROVIDE SPACE FOR THE
12217  047036  016666  000004  000002          MOV     4(SP),2(SP)     ;;INPUT NUMBER
12218  047044  010046                          MOV     R0,-(SP)        ;;PUSH R0 ON STACK
12219  047046  010146                          MOV     R1,-(SP)        ;;PUSH R1 ON STACK
12220  047050  010246                          MOV     R2,-(SP)        ;;PUSH R2 ON STACK
12221  047052  104411                  1$:     RDLIN                   ;;READ AN ASCIZ LINE
12222  047054  012600                          MOV     (SP)+,R0        ;;GET ADDRESS OF 1ST CHARACTER
12223  047056  010037  047162                  MOV     R0,5$           ;;AND SAVE IT
12224  047062  005001                          CLR     R1              ;;CLEAR DATA WORD
12225  047064  005002                          CLR     R2
12226  047066  112046                  2$:     MOVB    (R0)+,-(SP)     ;;PICKUP THIS CHARACTER
12227  047070  001420                          BEQ     3$              ;;IF ZERO GET OUT
12228  047072  122716  000060                  CMPB    #'0,(SP)        ;;MAKE SURE THIS CHARACTER
12229  047076  003026                          BGT     4$              ;;IS AN OCTAL DIGIT
12230  047100  122716  000067                  CMPB    #'7,(SP)
12231  047104  002423                          BLT     4$
12232  047106  006301                          ASL     R1              ;;*2
12233  047110  006102                          ROL     R2
12234  047112  006301                          ASL     R1              ;;*4
12235  047114  006102                          ROL     R2
12236  047116  006301                          ASL     R1              ;;*8
12237  047120  006102                          ROL     R2
12238  047122  042716  177770                  BIC     #^C7,(SP)       ;;STRIP THE ASCII JUNK
12239  047126  062601                          ADD     (SP)+,R1        ;;ADD IN THIS DIGIT
12240  047130  000756                          BR      2$              ;;LOOP
12241  047132  005726                  3$:     TST     (SP)+           ;;CLEAN TERMINATOR FROM STACK
12242  047134  010166  000012                  MOV     R1,12(SP)       ;;SAVE THE RESULT
12243  047140  010237  047172                  MOV     R2,$HIOCT
12244  047144  012602                          MOV     (SP)+,R2        ;;POP STACK INTO R2
12245  047146  012601                          MOV     (SP)+,R1        ;;POP STACK INTO R1
12246  047150  012600                          MOV     (SP)+,R0        ;;POP STACK INTO R0
12247  047152  000002                          RTI                     ;;RETURN
12248  047154  005726                  4$:     TST     (SP)+           ;;CLEAN PARTIAL FROM STACK
12249  047156  105010                          CLRB    (R0)            ;;SET A TERMINATOR
12250  047160  104401                          TYPE                    ;;TYPE UP THRU THE BAD CHAR.
12251  047162  000000                  5$:     .WORD   0
12252  047164  104401  001222                  TYPE    ,$QUES          ;;"?" "CR" & "LF"
12253  047170  000730                          BR      1$              ;;TRY AGAIN
12254  047172  000000                  $HIOCT: .WORD   0               ;;HIGH ORDER BITS GO HERE
```

```
                                .SBTTL  ERROR HANDLER ROUTINE

                                ;;*********************************************************************
                                ;*THIS ROUTINE WILL INCREMENT THE ERROR FLAG AND THE ERROR COUNT,
                                ;*SAVE THE ERROR ITEM NUMBER AND THE ADDRESS OF THE ERROR CALL
                                ;*AND GO TO $ERRTYP ON ERROR
                                ;*THE SWITCH OPTIONS PROVIDED BY THIS ROUTINE ARE:
                                ;*SW15=1         HALT ON ERROR
                                ;*SW13=1         INHIBIT ERROR TYPEOUTS
                                ;*SW10=1         BELL ON ERROR
                                ;*SW09=1         LOOP ON ERROR
                                ;*CALL
                                ;*      ERROR   N       ;;ERROR=EMT AND N=ERROR ITEM NUMBER

047174                          $ERROR:
047174  104407                          CKSWR                   ;;TEST FOR CHANGE IN SOFT-SWR

047176                          REGSAV:
047176  012737  177777  004734          MOV     #-1,@#ERFLG$    ;SET ERROR FLAG
047204                          REGSA1:
047204  105237  001103          7$:     INCB    $ERFLG          ;;SET THE ERROR FLAG
047210  001775                          BEQ     7$              ;;DON'T LET THE FLAG GO TO ZERO
047212  013777  001102  131722          MOV     $TSTNM,@DISPLAY ;;DISPLAY TEST NUMBER AND ERROR FLAG
047220  032777  002000  131712          BIT     #BIT10,@SWR     ;;BELL ON ERROR?
047226  001402                          BEQ     1$              ;;NO - SKIP
047230  104401  001216                  TYPE    ,$BELL          ;;RING BELL
047234  005237  001112          1$:     INC     $ERTTL          ;;COUNT THE NUMBER OF ERRORS
047240  011637  001116                  MOV     (SP),$ERRPC     ;;GET ADDRESS OF ERROR INSTRUCTION
047244  162737  000002  001116          SUB     #2,$ERRPC
047252  117737  131640  001114          MOVB    @$ERRPC,$ITEMB  ;;STRIP AND SAVE THE ERROR ITEM CODE
047260  032777  020000  131652          BIT     #BIT13,@SWR     ;;SKIP TYPEOUT IF SET
047266  001004                          BNE     20$             ;;SKIP TYPEOUTS
047270  004737  047344                  JSR     PC,$ERRTYP      ;;GO TO USER ERROR ROUTINE
047274  104401  001223                  TYPE    ,$CRLF
047300                          20$:
047300  005777  131634          2$:     TST     @SWR            ;;HALT ON ERROR
047304  100002                          BPL     3$              ;;SKIP IF CONTINUE
047306  000000                          HALT                    ;;HALT ON ERROR!
047310  104407                          CKSWR                   ;;TEST FOR CHANGE IN SOFT-SWR
047312  032777  001000  131620  3$:     BIT     #BIT09,@SWR     ;;LOOP ON ERROR SWITCH SET?
047320  001402                          BEQ     4$              ;;BR IF NO
047322  013716  001110                  MOV     $LPERR,(SP)     ;;FUDGE RETURN FOR LOOPING
047326  005737  001214          4$:     TST     $ESCAPE         ;;CHECK FOR AN ESCAPE ADDRESS
047332  001402                          BEQ     5$              ;;BR IF NONE
047334  013716  001214                  MOV     $ESCAPE,(SP)    ;;FUDGE RETURN ADDRESS FOR ESCAPE
047340                          5$:
047340  000002                          RTI                     ;;RETURN
```

```
12303
12304                                   ;*;**********************************
12305
12306                                   .SBTTL  ERROR MESSAGE TYPEOUT ROUTINE
12307
12308                                   ;*THIS ROUTINE USES THE "ITEM CONTROL BYTE" ($ITEMB) TO DETERMINE WHICH
12309                                   ;*ERROR IS TO BE REPORTED.  IT THEN OBTAINS, FROM THE "ERROR TABLE" ($ERRTB),
12310                                   ;*AND REPORTS THE APPROPRIATE INFORMATION CONCERNING THE ERROR.
12311                                   ;*IT IS A COPY OF THE $ERRTYP SUBROUTINE FROM SYSMAC.
12312                                   ;*WITH ONLY MINOR CHANGES
12313                                   ;*FIRST IF SWITCH 6 IS SET AND SWITCH 8 RESET THEN
12314                                   ;*ALL REGISTER CONTENTS WILL BE TYPED BEFOR REPORTING THE ERROR
12315                                   ;*SECOND IF THE CURRENT ERROR HAS THE SAME ITEM NUMBER
12316                                   ;*AS THE PREVIOUS ERROR THEN ONLY THE DATA WILL BE TYPED
12317                                   ;*AND NOT THE ERROR MESSAGE AND HEADER.
12318
12319  047342  000000                   PRITEM: 0                       ;PREVIOUS ITEM NO. LOCATION
12320
12321  047344  017746  131570           $ERRTYP: MOV    @SWR,-(SP)      ;GET SWITCH SETTING
12322  047350  042716  177277                   BIC    #^C500,(SP)     ;KEEP ONLY SWITCH 8 AND 6
12323  047354  022726  000100                   CMP    #SW06,(SP)+     ;IS 6 SET AND 8 RESET
12324  047360  001001                           BNE    1$              ;IF NOT BRANCH
12325  047362  000402                           BR     2$              ;BRANCH IF SW 6 IS SET AND 8 RESET
12326  047364  000137  050304           1$:     JMP    @#TYPERR        ;JUMP IF SW 8 IS SET
12327                                                                   ;OR IF SW 8 IS RESET AND SW 6 IS RESET
12328  047370                           2$:
12329
12330  047370  104401  047376                   TYPE   ,65$            ;;TYPE ASCIZ STRING
12331  047374  000406                           BR     64$             ;;GET OVER THE ASCIZ
12332                                   ;;65$:  .ASCIZ  <15><12>/RHWC =  /
12333  047412                           64$:
12334  047412  013746  002354                   MOV    @#WC,-(SP)      ;GET READY TO TYPE RHWC CONTENTS
12335  047416  104402                           TYPOC
12336
12337
12338  047420  104401  047426                   TYPE   ,67$            ;;TYPE ASCIZ STRING
12339  047424  000406                           BR     66$             ;;GET OVER THE ASCIZ
12340                                   ;;67$:  .ASCIZ  <15><12>/RHBA =  /
12341  047442                           66$:
12342  047442  013746  002356                   MOV    @#BA,-(SP)      ;GET READY TO TYPE RHBA CONTENTS
12343  047446  104402                           TYPOC
12344
12345
12346  047450  104401  047456                   TYPE   ,69$            ;;TYPE ASCIZ STRING
12347  047454  000406                           BR     68$             ;;GET OVER THE ASCIZ
12348                                   ;;69$:  .ASCIZ  <15><12>/RHCS2 =  /
12349  047472                           68$:
12350  047472  013746  002360                   MOV    @#CS2,-(SP)     ;GET READY TO TYPE RHCS2 CONTENTS
12351  047476  104402                           TYPOC
12352
12353
12354  047500  104401  047506                   TYPE   ,71$            ;;TYPE ASCIZ STRING
12355  047504  000406                           BR     70$             ;;GET OVER THE ASCIZ
12356                                   ;;71$:  .ASCIZ  <15><12>/RHCS1 =  /
12357  047522                           70$:
12358  047522  013746  002362                   MOV    @#CS1,-(SP)     ;GET READY TO TYPE RHCS1 CONTENTS
```

```
12359  047526  104402                          TYPOC
12360
12361
12362  047530  104401  047536                  TYPE    ,73$            ;;TYPE ASCIZ STRING
12363  047534  000406                          BR      72$             ;;GET OVER THE ASCIZ
12364                                  ;;73$:  .ASCIZ  <15><12>/RHDS1 =  /
12365  047552                          72$:
12366  047552  013746  002404                  MOV     @#DS1,-(SP)     ;GET READY TO TYPE RHDS1 CONTENTS
12367  047556  104402                          TYPOC
12368
12369
12370  047560  104401  047566                  TYPE    ,75$            ;;TYPE ASCIZ STRING
12371  047564  000406                          BR      74$             ;;GET OVER THE ASCIZ
12372                                  ;;75$:  .ASCIZ  <15><12>/RHER1 =  /
12373  047602                          74$:
12374  047602  013746  002364                  MOV     @#ER1,-(SP)     ;GET READY TO TYPE RHER1 CONTENTS
12375  047606  104402                          TYPOC
12376
12377
12378  047610  104401  047616                  TYPE    ,77$            ;;TYPE ASCIZ STRING
12379  047614  000406                          BR      76$             ;;GET OVER THE ASCIZ
12380                                  ;;77$:  .ASCIZ  <15><12>/RHER2 =  /
12381  047632                          76$:
12382  047632  013746  002370                  MOV     @#ER2,-(SP)     ;GET READY TO TYPE RHER2 CONTENTS
12383  047636  104402                          TYPOC
12384
12385
12386  047640  104401  047646                  TYPE    ,79$            ;;TYPE ASCIZ STRING
12387  047644  000406                          BR      78$             ;;GET OVER THE ASCIZ
12388                                  ;;79$:  .ASCIZ  <15><12>/RHER3 =  /
12389  047662                          78$:
12390  047662  013746  002376                  MOV     @#ER3,-(SP)     ;GET READY TO TYPE RHER3 CONTENTS
12391  047666  104402                          TYPOC
12392
12393
12394  047670  104401  047676                  TYPE    ,81$            ;;TYPE ASCIZ STRING
12395  047674  000406                          BR      80$             ;;GET OVER THE ASCIZ
12396                                  ;;81$:  .ASCIZ  <15><12>/RHDST =  /
12397  047712                          80$:
12398  047712  013746  002366                  MOV     @#DST,-(SP)     ;GET READY TO TYPE RHDST CONTENTS
12399  047716  104402                          TYPOC
12400
12401
12402  047720  104401  047726                  TYPE    ,83$            ;;TYPE ASCIZ STRING
12403  047724  000406                          BR      82$             ;;GET OVER THE ASCIZ
12404                                  ;;83$:  .ASCIZ  <15><12>/RHCA =  /
12405  047742                          82$:
12406  047742  013746  002374                  MOV     @#CA,-(SP)      ;GET READY TO TYPE RHCA CONTENTS
12407  047746  104402                          TYPOC
12408
12409
12410  047750  104401  047756                  TYPE    ,85$            ;;TYPE ASCIZ STRING
12411  047754  000406                          BR      84$             ;;GET OVER THE ASCIZ
12412                                  ;;85$:  .ASCIZ  <15><12>/RHAS =  /
12413  047772                          84$:
12414  047772  013746  002400                  MOV     @#AS,-(SP)      ;GET READY TO TYPE RHAS CONTENTS
```

```
12415  047776  104402                          TYPOC
12416
12417
12418  050000  104401  050006                  TYPE    ,87$            ;;TYPE ASCIZ STRING
12419  050004  000406                          BR      86$             ;;GET OVER THE ASCIZ
12420                                  ;;87$:   .ASCIZ  <15><12>/RHOF =  /
12421  050022                          86$:
12422  050022  013746  002372                  MOV     @#OF,-(SP)      ;GET READY TO TYPE RHOF CONTENTS
12423  050026  104402                          TYPOC
12424
12425
12426  050030  104401  050036                  TYPE    ,89$            ;;TYPE ASCIZ STRING
12427  050034  000406                          BR      88$             ;;GET OVER THE ASCIZ
12428                                  ;;89$:   .ASCIZ  <15><12>/RHMR =  /
12429  050052                          88$:
12430  050052  013746  002402                  MOV     @#MR,-(SP)      ;GET READY TO TYPE RHMR CONTENTS
12431  050056  104402                          TYPOC
12432
12433
12434  050060  104401  050066                  TYPE    ,91$            ;;TYPE ASCIZ STRING
12435  050064  000406                          BR      90$             ;;GET OVER THE ASCIZ
12436                                  ;;91$:   .ASCIZ  <15><12>/RHLA =  /
12437  050102                          90$:
12438  050102  013746  002420                  MOV     @#LA,-(SP)      ;GET READY TO TYPE RHLA CONTENTS
12439  050106  104402                          TYPOC
12440
12441
12442  050110  104401  050116                  TYPE    ,93$            ;;TYPE ASCIZ STRING
12443  050114  000406                          BR      92$             ;;GET OVER THE ASCIZ
12444                                  ;;93$:   .ASCIZ  <15><12>/RHCC =  /
12445  050132                          92$:
12446  050132  013746  002416                  MOV     @#CC,-(SP)      ;GET READY TO TYPE RHCC CONTENTS
12447  050136  104402                          TYPOC
12448
12449
12450  050140  104401  050146                  TYPE    ,95$            ;;TYPE ASCIZ STRING
12451  050144  000406                          BR      94$             ;;GET OVER THE ASCIZ
12452                                  ;;95$:   .ASCIZ  <15><12>/RHEC1 =  /
12453  050162                          94$:
12454  050162  013746  002412                  MOV     @#EC1,-(SP)     ;GET READY TO TYPE RHEC1 CONTENTS
12455  050166  104402                          TYPOC
12456
12457
12458  050170  104401  050176                  TYPE    ,97$            ;;TYPE ASCIZ STRING
12459  050174  000406                          BR      96$             ;;GET OVER THE ASCIZ
12460                                  ;;97$:   .ASCIZ  <15><12>/RHEC2 =  /
12461  050212                          96$:
12462  050212  013746  002414                  MOV     @#EC2,-(SP)     ;GET READY TO TYPE RHEC2 CONTENTS
12463  050216  104402                          TYPOC
12464
12465
12466  050220  104401  050226                  TYPE    ,99$            ;;TYPE ASCIZ STRING
12467  050224  000406                          BR      98$             ;;GET OVER THE ASCIZ
12468                                  ;;99$:   .ASCIZ  <15><12>/RHDT =  /
12469  050242                          98$:
12470  050242  013746  002406                  MOV     @#DT,-(SP)      ;GET READY TO TYPE RHDT CONTENTS
```

```
12471  050246  104402                        TYPOC
12472
12473
12474  050250  104401  050256                TYPE    ,101$           ;;TYPE ASCIZ STRING
12475  050254  000406                        BR      100$            ;;GET OVER THE ASCIZ
12476                                ;;101$: .ASCIZ  <15><12>/RHSN =  /
12477  050272                        100$:
12478  050272  013746  002410                MOV     @#SN,-(SP)      ;GET READY TO TYPE RHSN CONTENTS
12479  050276  104402                        TYPOC
12480
12481  050300  005037  047342                CLR     @#PRITEM        ;CLEAR PREVIOUS ERROR ITEM
12482  050304                        TYPERR:
12483  050304  104401  001223                TYPE    ,$CRLF          ;"CARRIAGE RETURN" & "LINE FEED"
12484  050310  010046                        MOV     R0,-(SP)        ;SAVE R0
12485  050312  005000                        CLR     R0              ;PICKUP THE ITEM INDEX
12486  050314  153700  001114                BISB    @#$ITEMB,R0
12487  050320  001004                        BNE     1$              ;IF ITEM NUMBER IS ZERO, JUST
12488                                                                ;TYPE THE PC OF THE ERROR
12489  050322  013746  001116                MOV     $ERRPC,-(SP)    ;SAVE $ERRPC FOR TYPEOUT
12490                                                                ;ERROR ADDRESS
12491  050326  104402                        TYPOC                   ;GO TYPE--OCTAL ASCII(ALL DIGITS)
12492  050330  000454                        BR      10$             ;GET OUT
12493  050332  005300                1$:     DEC     R0              ;ADJUST THE INDEX SO THAT IT WILL
12494  050334  006300                        ASL     R0              ;       WORK FOR THE ERROR TABLE
12495  050336  006300                        ASL     R0
12496  050340  006300                        ASL     R0
12497  050342  062700  001226                ADD     #$ERRTB,R0      ;FORM TABLE POINTER
12498  050346  020037  047342                CMP     R0,@#PRITEM     ;WAS PREVIOUS ERROR SAME
12499  050352  001002                        BNE     13$             ;BRANCH IF NOT
12500  050354  022020                        CMP     (R0)+,(R0)+     ;POP R0 OVER EM AND DH
12501  050356  000420                        BR      5$
12502  050360  010037  047342        13$:    MOV     R0,@#PRITEM     ;SAVE NEW ERROR ITEM
12503  050364  012037  050374                MOV     (R0)+,2$        ;PICKUP "ERROR MESSAGE" POINTER
12504  050370  001404                        BEQ     3$              ;SKIP TYPEOUT IF NO POINTER
12505  050372  104401                        TYPE                    ;TYPE THE "ERROR MESSAGE"
12506  050374  000000                2$:     .WORD   0               ;"ERROR MESSAGE" POINTER GOES HERE
12507  050376  104401  001223                TYPE    ,$CRLF          ;"CARRIAGE RETURN" & "LINE FEED"
12508  050402  012037  050412        3$:     MOV     (R0)+,4$        ;PICKUP "DATA HEADER" POINTER
12509  050406  001404                        BEQ     5$              ;SKIP TYPEOUT IF 0
12510  050410  104401                        TYPE                    ;TYPE THE "DATA HEADER"
12511  050412  000000                4$:     .WORD   0               ;"DATA HEADER" POINTER GOES HERE
12512  050414  104401  001223                TYPE    ,$CRLF          ;"CARRIAGE RETURN" & "LINE FEED"
12513  050420  010146                5$:     MOV     R1,-(SP)        ;SAVE R1
12514  050422  012001                        MOV     (R0)+,R1        ;PICKUP "DATA TABLE" POINTER
12515  050424  001415                        BEQ     9$              ;BR IF NO DATA TO BE TYPED
12516  050426  012000                        MOV     (R0)+,R0        ;PICKUP "DATA FORMAT" POINTER
12517  050430  105720                6$:     TSTB    (R0)+           ;"OCTAL" OR "DECIMAL"
12518  050432  001003                        BNE     7$              ;BR IF DECIMAL
12519  050434  013146                        MOV     @(R1)+,-(SP)    ;SAVE @(R1)+ FOR TYPEOUT
12520  050436  104402                        TYPOC                   ;GO TYPE--OCTAL ASCII(ALL DIGITS)
12521  050440  000402                        BR      8$
12522  050442                        7$:
12523  050442  013146                        MOV     @(R1)+,-(SP)    ;SAVE @(R1)+ FOR TYPEOUT
12524  050444  104405                        TYPDS                   ;GO TYPE--DECIMAL ASCII WITH SIGN
12525  050446  005711                8$:     TST     (R1)            ;IS THERE ANOTHER NUMBER?
12526  050450  001403                        BEQ     9$              ;BR IF NO
```

```
12527  050452  104401  050466              TYPE    ,11$            ;TYPE TWO(2) SPACES
12528  050456  000764                      BR      6$              ;LOOP
12529
12530  050460  012601                9$:   MOV     (SP)+,R1        ;RESTORE R1
12531  050462  012600                10$:  MOV     (SP)+,R0        ;"CARRIAGE RETURN" & "LINE FEED"
12532  050464  000207                      RTS     PC              ;RETURN
12533  050466  020040     000        11$:  .ASCIZ  /  /            ;TWO(2) SPACES
12534          050472                      .EVEN
12535
```

```
12536
12537                                   .SBTTL  BINARY TO OCTAL (ASCII) AND TYPE
12538
12539                                   ;;**********************************************************************
12540                                   ;*THIS ROUTINE IS USED TO CHANGE A 16-BIT BINARY NUMBER TO A 6-DIGIT
12541                                   ;*OCTAL (ASCII) NUMBER AND TYPE IT.
12542                                   ;*$TYPOS---ENTER HERE TO SETUP SUPPRESS ZEROS AND NUMBER OF DIGITS TO TYPE
12543                                   ;*CALL:
12544                                   ;*      MOV     NUM,-(SP)       ;;NUMBER TO BE TYPED
12545                                   ;*      TYPOS                   ;;CALL FOR TYPEOUT
12546                                   ;*      .BYTE   N               ;;N=1 TO 6 FOR NUMBER OF DIGITS TO TYPE
12547                                   ;*      .BYTE   M               ;;M=1 OR 0
12548                                   ;*                                      ;;1=TYPE LEADING ZEROS
12549                                   ;*                                      ;;0=SUPPRESS LEADING ZEROS
12550                                   ;*
12551                                   ;*$TYPON----ENTER HERE TO TYPE OUT WITH THE SAME PARAMETERS AS THE LAST
12552                                   ;*$TYPOS OR $TYPOC
12553                                   ;*CALL:
12554                                   ;*      MOV     NUM,-(SP)       ;;NUMBER TO BE TYPED
12555                                   ;*      TYPON                   ;;CALL FOR TYPEOUT
12556                                   ;*
12557                                   ;*$TYPOC---ENTER HERE FOR TYPEOUT OF A 16 BIT NUMBER
12558                                   ;*CALL:
12559                                   ;*      MOV     NUM,-(SP)       ;;NUMBER TO BE TYPED
12560                                   ;*      TYPOC                   ;;CALL FOR TYPEOUT
12561
12562  050472  017646  000000          $TYPOS: MOV     @(SP),-(SP)     ;;PICKUP THE MODE
12563  050476  116637  000001  050715          MOVB    1(SP),$OFILL    ;;LOAD ZERO FILL SWITCH
12564  050504  112637  050717                  MOVB    (SP)+,$OMODE+1  ;;NUMBER OF DIGITS TO TYPE
12565  050510  062716  000002                  ADD     #2,(SP)         ;;ADJUST RETURN ADDRESS
12566  050514  000406                          BR      $TYPON
12567  050516  112737  000001  050715  $TYPOC: MOVB    #1,$OFILL       ;;SET THE ZERO FILL SWITCH
12568  050524  112737  000006  050717          MOVB    #6,$OMODE+1     ;;SET FOR SIX(6) DIGITS
12569  050532  112737  000005  050714  $TYPON: MOVB    #5,$OCNT        ;;SET THE ITERATION COUNT
12570  050540  010346                          MOV     R3,-(SP)        ;;SAVE R3
12571  050542  010446                          MOV     R4,-(SP)        ;;SAVE R4
12572  050544  010546                          MOV     R5,-(SP)        ;;SAVE R5
12573  050546  113704  050717                  MOVB    $OMODE+1,R4     ;;GET THE NUMBER OF DIGITS TO TYPE
12574  050552  005404                          NEG     R4
12575  050554  062704  000006                  ADD     #6,R4           ;;SUBTRACT IT FOR MAX. ALLOWED
12576  050560  110437  050716                  MOVB    R4,$OMODE       ;;SAVE IT FOR USE
12577  050564  113704  050715                  MOVB    $OFILL,R4       ;;GET THE ZERO FILL SWITCH
12578  050570  016605  000012                  MOV     12(SP),R5       ;;PICKUP THE INPUT NUMBER
12579  050574  005003                          CLR     R3              ;;CLEAR THE OUTPUT WORD
12580  050576  006105                  1$:     ROL     R5              ;;ROTATE MSB INTO "C"
12581  050600  000404                          BR      3$              ;;GO DO MSB
12582  050602  006105                  2$:     ROL     R5              ;;FORM THIS DIGIT
12583  050604  006105                          ROL     R5
12584  050606  006105                          ROL     R5
12585  050610  010503                          MOV     R5,R3
12586  050612  006103                  3$:     ROL     R3              ;;GET LSB OF THIS DIGIT
12587  050614  105337  050716                  DECB    $OMODE          ;;TYPE THIS DIGIT?
12588  050620  100016                          BPL     7$              ;;BR IF NO
12589  050622  042703  177770                  BIC     #177770,R3      ;;GET RID OF JUNK
12590  050626  001002                          BNE     4$              ;;TEST FOR 0
12591  050630  005704                          TST     R4              ;;SUPPRESS THIS 0?
```

```
12592  050632  001403                        BEQ     5$              ;;BR IF YES
12593  050634  005204                4$:     INC     R4              ;;DON'T SUPPRESS ANYMORE 0'S
12594  050636  052703  000060              BIS     #'0,R3          ;;MAKE THIS DIGIT ASCII
12595  050642  052703  000040        5$:   BIS     #' ,R3          ;;MAKE ASCII IF NOT ALREADY
12596  050646  110337  050712              MOVB    R3,8$           ;;SAVE FOR TYPING
12597  050652  104401  050712              TYPE    ,8$             ;;GO TYPE THIS DIGIT
12598  050656  105337  050714        7$:   DECB    $OCNT           ;;COUNT BY 1
12599  050662  003347                        BGT     2$              ;;BR IF MORE TO DO
12600  050664  002402                        BLT     6$              ;;BR IF DONE
12601  050666  005204                        INC     R4              ;;INSURE LAST DIGIT ISN'T A BLANK
12602  050670  000744                        BR      2$              ;;GO DO THE LAST DIGIT
12603  050672  012605                6$:     MOV     (SP)+,R5        ;;RESTORE R5
12604  050674  012604                        MOV     (SP)+,R4        ;;RESTORE R4
12605  050676  012603                        MOV     (SP)+,R3        ;;RESTORE R3
12606  050700  016666  000002  000004       MOV     2(SP),4(SP)     ;;SET THE STACK FOR RETURNING
12607  050706  012616                        MOV     (SP)+,(SP)
12608  050710  000002                        RTI                     ;;RETURN
12609  050712     000                8$:     .BYTE   0               ;;STORAGE FOR ASCII DIGIT
12610  050713     000                        .BYTE   0               ;;TERMINATOR FOR TYPE ROUTINE
12611  050714     000             $OCNT:     .BYTE   0               ;;OCTAL DIGIT COUNTER
12612  050715     000             $OFILL:    .BYTE   0               ;;ZERO FILL SWITCH
12613  050716  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.

050720  010046                  $TRAP:  MOV     R0,-(SP)        ;;SAVE R0
050722  016600  000002                  MOV     2(SP),R0        ;;GET TRAP ADDRESS
050726  005740                          TST     -(R0)           ;;BACKUP BY 2
050730  111000                          MOVB    (R0),R0         ;;GET RIGHT BYTE OF TRAP
050732  006300                          ASL     R0              ;;POSITION FOR INDEXING
050734  016000  050754                  MOV     $TRPAD(R0),R0   ;;INDEX TO TABLE
050740  000200                          RTS     R0              ;;GO TO ROUTINE

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

050742  011646                  $TRAP2: MOV     (SP),-(SP)      ;;MOVE THE PC DOWN
050744  016666  000004  000002          MOV     4(SP),2(SP)     ;;MOVE THE PSW DOWN
050752  000002                          RTI                     ;;RESTORE THE PSW

                                        .SBTTL  TRAP TABLE

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

                                        ;       ROUTINE
                                        ;       -------
050754  050742                  $TRPAD: .WORD   $TRAP2
050756  045474                          $TYPE   ;;CALL=TYPE     TRAP+1(104401)  TTY TYPEOUT ROUTINE
050760  050516                          $TYPOC  ;;CALL=TYPOC    TRAP+2(104402)  TYPE OCTAL NUMBER (WITH LEADING ZEROS)
050762  050472                          $TYPOS  ;;CALL=TYPOS    TRAP+3(104403)  TYPE OCTAL NUMBER (NO LEADING ZEROS)
050764  050532                          $TYPON  ;;CALL=TYPON    TRAP+4(104404)  TYPE OCTAL NUMBER (AS PER LAST CALL)
050766  045250                          $TYPDS  ;;CALL=TYPDS    TRAP+5(104405)  TYPE DECIMAL NUMBER (WITH SIGN)

050770  046312                          $GTSWR  ;;CALL=GTSWR    TRAP+6(104406)  GET SOFT-SWR SETTING

050772  046222                          $CKSWR  ;;CALL=CKSWR    TRAP+7(104407)  TEST FOR CHANGE IN SOFT-SWR
050774  046564                          $RDCHR  ;;CALL=RDCHR    TRAP+10(104410) TTY TYPEIN CHARACTER ROUTINE
050776  046654                          $RDLIN  ;;CALL=RDLIN    TRAP+11(104411) TTY TYPEIN STRING ROUTINE
051000  047034                          $RDOCT  ;;CALL=RDOCT    TRAP+12(104412) READ AN OCTAL NUMBER FROM TTY
051002  042224                          WAIT.T  ;;CALL=WAT      TRAP+13(104413) DONT ADD ABOVE THIS TRAP
```

```
12659                                   .SBTTL  POWER DOWN AND UP ROUTINES
12660
12661                                   ;;*****************************************************************
12662                                   ;POWER DOWN ROUTINE
12663  051004  012737  051150  000024  $PWRDN: MOV     #$ILLUP,@#PWRVEC ;;SET FOR FAST UP
12664  051012  012737  000340  000026          MOV     #340,@#PWRVEC+2 ;;PRIO:7
12665  051020  010046                         MOV     R0,-(SP)        ;;PUSH R0 ON STACK
12666  051022  010146                         MOV     R1,-(SP)        ;;PUSH R1 ON STACK
12667  051024  010246                         MOV     R2,-(SP)        ;;PUSH R2 ON STACK
12668  051026  010346                         MOV     R3,-(SP)        ;;PUSH R3 ON STACK
12669  051030  010446                         MOV     R4,-(SP)        ;;PUSH R4 ON STACK
12670  051032  010546                         MOV     R5,-(SP)        ;;PUSH R5 ON STACK
12671  051034  017746  130100                 MOV     @SWR,-(SP)      ;;PUSH @SWR ON STACK
12672  051040  010637  051154                 MOV     SP,$SAVR6       ;;SAVE SP
12673  051044  012737  051056  000024         MOV     #$PWRUP,@#PWRVEC ;;SET UP VECTOR
12674  051052  000000                         HALT
12675  051054  000776                         BR      .-2             ;;HANG UP
12676
12677                                   ;;*****************************************************************
12678                                   ;POWER UP ROUTINE
12679  051056  012737  051150  000024  $PWRUP: MOV     #$ILLUP,@#PWRVEC ;;SET FOR FAST DOWN
12680  051064  013706  051154                 MOV     $SAVR6,SP       ;;GET SP
12681  051070  005037  051154                 CLR     $SAVR6          ;;WAIT LOOP FOR THE TTY
12682  051074  005237  051154          1$:    INC     $SAVR6          ;;WAIT FOR THE INC
12683  051100  001375                         BNE     1$              ;;OF  WORD
12684  051102  012677  130032                 MOV     (SP)+,@SWR      ;;POP STACK INTO @SWR
12685  051106  012605                         MOV     (SP)+,R5        ;;POP STACK INTO R5
12686  051110  012604                         MOV     (SP)+,R4        ;;POP STACK INTO R4
12687  051112  012603                         MOV     (SP)+,R3        ;;POP STACK INTO R3
12688  051114  012602                         MOV     (SP)+,R2        ;;POP STACK INTO R2
12689  051116  012601                         MOV     (SP)+,R1        ;;POP STACK INTO R1
12690  051120  012600                         MOV     (SP)+,R0        ;;POP STACK INTO R0
12691  051122  012737  051004  000024         MOV     #$PWRDN,@#PWRVEC ;;SET UP THE POWER DOWN VECTOR
12692  051130  012737  000340  000026         MOV     #340,@#PWRVEC+2 ;;PRIO:7
12693  051136  104401                         TYPE                    ;REPORT THE POWER FAILURE
12694  051140  051156          $PWRMG: .WORD   $POWER          ;;POWER FAIL MESSAGE POINTER
12695  051142  012716                  MOV     (PC)+,(SP)      ;;RESTART AT BEGIN
12696  051144  005012          $PWRAD: .WORD   BEGIN           ;;RESTART ADDRESS
12697  051146  000002                  RTI
12698  051150  000000          $ILLUP: HALT                    ;;THE POWER UP SEQUENCE WAS STARTED
12699  051152  000776                  BR      .-2             ;; BEFORE THE POWER DOWN WAS COMPLETE
12700  051154  000000          $SAVR6: 0                       ;;PUT THE SP HERE
12701  051156  005015  047520  042527  $POWER: .ASCIZ  <15><12>''POWER''
12702  051164  000122
12703                                   .EVEN
```

```
12704                                   ;*************************************
12705                                   ;*
12706                                   ;*ERROR AND MESSAGE TABLE CONDIMENTS
12707                                   ;*
12708                                   ;*************************************
12709
12710
12711
12712
12713  051166  050122  032060  042040  EM1:    .ASCIZ  /RP04 DID NOT INTERRUPT/
12714  051174  042111  047040  052117
12715  051202  044440  052116  051105
12716  051210  052522  052120     000
12717  051215     111  052116  051105  EM2:    .ASCIZ  /INTERRUPT ENABLE BIT DOWN BUT EXPECTED BIT DID NOT SET/
12718  051222  052522  052120  042440
12719  051230  040516  046102  020105
12720  051236  044502  020124  047504
12721  051244  047127  041040  052125
12722  051252  042440  050130  041505
12723  051260  042524  020104  044502
12724  051266  020124  044504  020104
12725  051274  047516  020124  042523
12726  051302  000124
12727  051304  050122  032060  042040  EM3:    .ASCIZ  /RP04 DID NOT INTERRUPT WHEN EXPECTED BIT DID SET/
12728  051312  042111  047040  052117
12729  051320  044440  052116  051105
12730  051326  052522  052120  053440
12731  051334  042510  020116  054105
12732  051342  042520  052103  042105
12733  051350  041040  052111  042040
12734  051356  042111  051440  052105
12735  051364     000
12736  051365     105  050130  041505  EM4:    .ASCIZ  /EXPECTED BIT DID SET BUT TIME IS IN ERROR (TIME IN 10 MICROSEC, DECIMAL
12737  051372  042524  020104  044502
12738  051400  020124  044504  020104
12739  051406  042523  020124  052502
12740  051414  020124  044524  042515
12741  051422  044440  020123  047111
12742  051430  042440  051122  051117
12743  051436  024040  044524  042515
12744  051444  044440  020116  030061
12745  051452  046440  041511  047522
12746  051460  042523  026103  042040
12747  051466  041505  046511  046101
12748  051474  000051
12749  051476  044122  051501  042040  EM5:    .ASCIZ  /RHAS DOES NOT CLEAR BY MOVING IN ALL ONES/
12750  051504  042517  020123  047516
12751  051512  020124  046103  040505
12752  051520  020122  054502  046440
12753  051526  053117  047111  020107
12754  051534  047111  040440  046114
12755  051542  047440  042516  000123
12756  051550  047514  042101  047111  EM6:    .ASCIZ  /LOADING RHER1 FOR ALL UNITS DID NOT SET ANY RHAS BITS/
12757  051556  020107  044122  051105
12758  051564  020061  047506  020122
12759  051572  046101  020114  047125
```

```
12760  051600  052111  020123  044504
12761  051606  020104  047516  020124
12762  051614  042523  020124  047101
12763  051622  020131  044122  051501
12764  051630  041040  052111  000123
12765  051636  047516  020116  054105  EM7:    .ASCIZ  /NON EXISTENT REGISTER, PROGRAM ABORTED./
12766  051644  051511  042524  052116
12767  051652  051040  043505  051511
12768  051660  042524  026122  050040
12769  051666  047522  051107  046501
12770  051674  040440  047502  052122
12771  051702  042105  000056
12772
12773  051706  052123  050117  042520  EM10:   .ASCIZ  /STOPPED DRIVE HAS MOL BIT IN RHDS1 SET/
12774  051714  020104  051104  053111
12775  051722  020105  040510  020123
12776  051730  047515  020114  044502
12777  051736  020124  047111  051040
12778  051744  042110  030523  051440
12779  051752  052105     000
12780  051755     127  052111  020110  EM11:   .ASCIZ  /WITH SPINDLE POWERED DOWN RHCS2 SHOULD ONLY HAVE UNIT NO. AND IR SET/
12781  051762  050123  047111  046104
12782  051770  020105  047520  042527
12783  051776  042522  020104  047504
12784  052004  047127  051040  041510
12785  052012  031123  051440  047510
12786  052020  046125  020104  047117
12787  052026  054514  044040  053101
12788  052034  020105  047125  052111
12789  052042  047040  027117  040440
12790  052050  042116  044440  020122
12791  052056  042523  000124
12792  052062  043101  042524  020122  EM12:   .ASCIZ  /AFTER SPINDLE POWERED UP, NO PACK ACKN. RHDS1 SHOULD HAVE MOL=1, VV=0/
12793  052070  050123  047111  046104
12794  052076  020105  047520  042527
12795  052104  042522  020104  050125
12796  052112  020054  047516  050040
12797  052120  041501  020113  041501
12798  052126  047113  020056  044122
12799  052134  051504  020061  044123
12800  052142  052517  042114  044040
12801  052150  053101  020105  047515
12802  052156  036514  026061  053040
12803  052164  036526  000060
12804  052170  044527  044124  051440  EM13:   .ASCIZ  /WITH SPINDLE POWERED UP, NO INTIALIZE, RHCS1 SHOULD HAVE GO=0, DVA=1, R
12805  052176  044520  042116  042514
12806  052204  050040  053517  051105
12807  052212  042105  052440  026120
12808  052220  047040  020117  047111
12809  052226  044524  046101  055111
12810  052234  026105  051040  041510
12811  052242  030523  051440  047510
12812  052250  046125  020104  040510
12813  052256  042526  043440  036517
12814  052264  026060  042040  040526
12815  052272  030475  020054  042122
```

```
12816  052300  036531  026061  044440
12817  052306  036505  000060
12818  052312  043101  042524  020122  EM14:   .ASCIZ  /AFTER SPINDLE POWERED UP, RHCC SHOULD BE=0/
12819  052320  050123  047111  046104
12820  052326  020105  047520  042527
12821  052334  042522  020104  050125
12822  052342  020054  044122  041503
12823  052350  051440  047510  046125
12824  052356  020104  042502  030075
12825  052364     000
12826  052365     120  041501  020113  EM15:   .ASCII  /PACK ACKNOWLEDGE COMMAND CAUSED AN ERROR/<15><12>
12827  052372  041501  047113  053517
12828  052400  042514  043504  020105
12829  052406  047503  046515  047101
12830  052414  020104  040503  051525
12831  052422  042105  040440  020116
12832  052430  051105  047522  006522
12833  052436     012
12834  052437     107  047517  020104          .ASCIZ  /GOOD DATA IS BEFORE COMMAND, REC DATA IS AFTER COMMAND/
12835  052444  040504  040524  044440
12836  052452  020123  042502  047506
12837  052460  042522  041440  046517
12838  052466  040515  042116  020054
12839  052474  042522  020103  040504
12840  052502  040524  044440  020123
12841  052510  043101  042524  020122
12842  052516  047503  046515  047101
12843  052524  000104
12844  052526  047516  047455  020120  EM16:   .ASCII  /NO-OP COMMAND CAUSED AN ERROR/<15><12>
12845  052534  047503  046515  047101
12846  052542  020104  040503  051525
12847  052550  042105  040440  020116
12848  052556  051105  047522  006522
12849  052564     012
12850  052565     107  047517  020104          .ASCIZ  /GOOD DATA IS BEFORE COMMAND, REC DATA IS AFTER COMMAND/
12851  052572  040504  040524  044440
12852  052600  020123  042502  047506
12853  052606  042522  041440  046517
12854  052614  040515  042116  020054
12855  052622  042522  020103  040504
12856  052630  040524  044440  020123
12857  052636  043101  042524  020122
12858  052644  047503  046515  047101
12859  052652  000104
12860  052654  051104  053111  020105  EM17:   .ASCII  /DRIVE CLEAR COMMAND CAUSED AN ERROR/<15><12>
12861  052662  046103  040505  020122
12862  052670  047503  046515  047101
12863  052676  020104  040503  051525
12864  052704  042105  040440  020116
12865  052712  051105  047522  006522
12866  052720     012
12867  052721     107  047517  020104          .ASCIZ  /GOOD DATA GIVES SHOULD BE, REC DATA GIVES AFTER COMMAND/
12868  052726  040504  040524  043440
12869  052734  053111  051505  051440
12870  052742  047510  046125  020104
12871  052750  042502  020054  042522
```

```
12872  052756  020103  040504  040524
12873  052764  043440  053111  051505
12874  052772  040440  052106  051105
12875  053000  041440  046517  040515
12876  053006  042116     000
12877
12878  053011     122  040505  026504  EM20:   .ASCII  /READ-IN COMMAND CAUSED AN ERROR/<15><12>
12879  053016  047111  041440  046517
12880  053024  040515  042116  041440
12881  053032  052501  042523  020104
12882  053040  047101  042440  051122
12883  053046  051117  005015
12884  053052  047507  042117  042040          .ASCIZ  /GOOD DATA GIVES SHOULD BE, REC DATA GIVES REG. CONT. AFTER COMMAND/
12885  053060  052101  020101  044507
12886  053066  042526  020123  044123
12887  053074  052517  042114  041040
12888  053102  026105  051040  041505
12889  053110  042040  052101  020101
12890  053116  044507  042526  020123
12891  053124  042522  027107  041440
12892  053132  047117  027124  040440
12893  053140  052106  051105  041440
12894  053146  046517  040515  042116
12895  053154     000
12896  053155     122  041510  030523  EM21:   .ASCIZ  /RHCS1 CONTENTS DURING COMMAND WAS IN ERROR/
12897  053162  041440  047117  042524
12898  053170  052116  020123  052504
12899  053176  044522  043516  041440
12900  053204  046517  040515  042116
12901  053212  053440  051501  044440
12902  053220  020116  051105  047522
12903  053226  000122
12904  053230  044122  051504  020061  EM22:   .ASCIZ  /RHDS1 CONTENTS DURING COMMAND WAS IN ERROR/
12905  053236  047503  052116  047105
12906  053244  051524  042040  051125
12907  053252  047111  020107  047503
12908  053260  046515  047101  020104
12909  053266  040527  020123  047111
12910  053274  042440  051122  051117
12911  053302     000
12912  053303     125  046116  040517  EM23:   .ASCII  /UNLOAD COMMAND CAUSED AN ERROR/<15><12>
12913  053310  020104  047503  046515
12914  053316  047101  020104  040503
12915  053324  051525  042105  040440
12916  053332  020116  051105  047522
12917  053340  006522     012
12918  053343     107  047517  020104          .ASCIZ  /GOOD DATA GIVES SHOULD BE, REC DATA GIVES REG. CONT. AFTER COMMAND/
12919  053350  040504  040524  043440
12920  053356  053111  051505  051440
12921  053364  047510  046125  020104
12922  053372  042502  020054  042522
12923  053400  020103  040504  040524
12924  053406  043440  053111  051505
12925  053414  051040  043505  020056
12926  053422  047503  052116  020056
12927  053430  043101  042524  020122
```

```
12928  053436  047503  046515  047101
12929  053444  000104
12930  053446  043117  051506  052105  EM24:   .ASCII  /OFFSET COMMAND CAUSED AN ERROR/<15><12>
12931  053454  041440  046517  040515
12932  053462  042116  041440  052501
12933  053470  042523  020104  047101
12934  053476  042440  051122  051117
12935  053504  005015
12936  053506  047507  042117  042040          .ASCIZ  /GOOD DATA GIVES SHOULD BE, REC DATA GIVES REG. CONT. AFTER COMMAND/
12937  053514  052101  020101  044507
12938  053522  042526  020123  044123
12939  053530  052517  042114  041040
12940  053536  026105  051040  041505
12941  053544  042040  052101  020101
12942  053552  044507  042526  020123
12943  053560  042522  027107  041440
12944  053566  047117  027124  040440
12945  053574  052106  051105  041440
12946  053602  046517  040515  042116
12947  053610     000
12948  053611     122  052105  051125  EM25:   .ASCII  /RETURN TO CENTER LINE COMMAND CAUSED AN ERROR/<15><12>
12949  053616  020116  047524  041440
12950  053624  047105  042524  020122
12951  053632  044514  042516  041440
12952  053640  046517  040515  042116
12953  053646  041440  052501  042523
12954  053654  020104  047101  042440
12955  053662  051122  051117  005015
12956  053670  047507  042117  042040          .ASCIZ  /GOOD DATA GIVES SHOULD BE, REC DATA GIVES REG. CONT. AFTER COMMAND/
12957  053676  052101  020101  044507
12958  053704  042526  020123  044123
12959  053712  052517  042114  041040
12960  053720  026105  051040  041505
12961  053726  042040  052101  020101
12962  053734  044507  042526  020123
12963  053742  042522  027107  041440
12964  053750  047117  027124  040440
12965  053756  052106  051105  041440
12966  053764  046517  040515  042116
12967  053772     000
12968  053773     065  030060  047440  EM26:   .ASCIZ  /500 OFFSET COMMANDS ONE AFTER THE OTHER CAUSED AN ERROR/
12969  054000  043106  042523  020124
12970  054006  047503  046515  047101
12971  054014  051504  047440  042516
12972  054022  040440  052106  051105
12973  054030  052040  042510  047440
12974  054036  044124  051105  041440
12975  054044  052501  042523  020104
12976  054052  047101  042440  051122
12977  054060  051117     000
12978  054063     127  044522  042524  EM27:   .ASCII  /WRITE HEADER AND DATA CAUSED IMPROPER REGISTER CHANGE/<15><12>
12979  054070  044040  040505  042504
12980  054076  020122  047101  020104
12981  054104  040504  040524  041440
12982  054112  052501  042523  020104
12983  054120  046511  051120  050117
```

```
12984  054126  051105  051040  043505
12985  054134  051511  042524  020122
12986  054142  044103  047101  042507
12987  054150  005015
12988  054152  047507  042117  042040          .ASCII  /GOOD DATA GIVES WHAT SHOULD BE THERE/<15><12>
12989  054160  052101  020101  044507
12990  054166  042526  020123  044127
12991  054174  052101  051440  047510
12992  054202  046125  020104  042502
12993  054210  052040  042510  042522
12994  054216  005015
12995  054220  042522  042503  053111          .ASCIZ  /RECEIVED DATA GIVES WHAT WAS THERE AFTER COMMAND/
12996  054226  042105  042040  052101
12997  054234  020101  044507  042526
12998  054242  020123  044127  052101
12999  054250  053440  051501  052040
13000  054256  042510  042522  040440
13001  054264  052106  051105  041440
13002  054272  046517  040515  042116
13003  054300     000
13004
13005  054301     127  044522  042524  EM30:   .ASCIZ  /WRITE HEADER AND DATA CHANGED WRITE FROM BUFFER/
13006  054306  044040  040505  042504
13007  054314  020122  047101  020104
13008  054322  040504  040524  041440
13009  054330  040510  043516  042105
13010  054336  053440  044522  042524
13011  054344  043040  047522  020115
13012  054352  052502  043106  051105
13013  054360     000
13014  054361     122  040505  020104  EM31:   .ASCII  /READ HEADER AND DATA CAUSED IMPROPER REGISTER CHANGE/<15><12>
13015  054366  042510  042101  051105
13016  054374  040440  042116  042040
13017  054402  052101  020101  040503
13018  054410  051525  042105  044440
13019  054416  050115  047522  042520
13020  054424  020122  042522  044507
13021  054432  052123  051105  041440
13022  054440  040510  043516  006505
13023  054446     012
13024  054447     107  047517  020104          .ASCII  /GOOD DATA GIVES WHAT SHOULD BE THERE/<15><12>
13025  054454  040504  040524  043440
13026  054462  053111  051505  053440
13027  054470  040510  020124  044123
13028  054476  052517  042114  041040
13029  054504  020105  044124  051105
13030  054512  006505     012
13031  054515     122  041505  044505          .ASCIZ  /RECEIVED DATA GIVES WHAT WAS THERE AFTER COMMAND/
13032  054522  042526  020104  040504
13033  054530  040524  043440  053111
13034  054536  051505  053440  040510
13035  054544  020124  040527  020123
13036  054552  044124  051105  020105
13037  054560  043101  042524  020122
13038  054566  047503  046515  047101
13039  054574  000104
```

```
13040  054576  051127  052111  020105  EM32:   .ASCIZ  /WRITE HEADER DATA, FOLLOWED BY READ HEADER AND DATA, CAUSED DATA ERROR/
13041  054604  042510  042101  051105
13042  054612  042040  052101  026101
13043  054620  043040  046117  047514
13044  054626  042527  020104  054502
13045  054634  051040  040505  020104
13046  054642  042510  042101  051105
13047  054650  040440  042116  042040
13048  054656  052101  026101  041440
13049  054664  052501  042523  020104
13050  054672  040504  040524  042440
13051  054700  051122  051117     000
13052  054705     122  040505  020104  EM33:   .ASCII  /READ DATA CAUSED IMPROPER REGISTER CHANGE/<15><12>
13053  054712  040504  040524  041440
13054  054720  052501  042523  020104
13055  054726  046511  051120  050117
13056  054734  051105  051040  043505
13057  054742  051511  042524  020122
13058  054750  044103  047101  042507
13059  054756  005015
13060  054760  047507  042117  042040          .ASCII  /GOOD DATA GIVES WHAT SHOULD BE THERE/<15><12>
13061  054766  052101  020101  044507
13062  054774  042526  020123  044127
13063  055002  052101  051440  047510
13064  055010  046125  020104  042502
13065  055016  052040  042510  042522
13066  055024  005015
13067  055026  042522  042503  053111          .ASCIZ  /RECEIVED DATA GIVES WHAT WAS THERE AFTER COMMAND/
13068  055034  042105  042040  052101
13069  055042  020101  044507  042526
13070  055050  020123  044127  052101
13071  055056  053440  051501  052040
13072  055064  042510  042522  040440
13073  055072  052106  051105  041440
13074  055100  046517  040515  042116
13075  055106     000
13076  055107     122  040505  020104  EM34:   .ASCIZ  /READ DATA INCORRECT/
13077  055114  040504  040524  044440
13078  055122  041516  051117  042522
13079  055130  052103     000
13080  055133     127  044522  042524  EM35:   .ASCII  /WRITE DATA COMMAND CAUSED IMPROPER REGISTER CHANGE/<15><12>
13081  055140  042040  052101  020101
13082  055146  047503  046515  047101
13083  055154  020104  040503  051525
13084  055162  042105  044440  050115
13085  055170  047522  042520  020122
13086  055176  042522  044507  052123
13087  055204  051105  041440  040510
13088  055212  043516  006505     012
13089  055217     107  047517  020104          .ASCII  /GOOD DATA GIVES WHAT SHOULD BE THERE/<15><12>
13090  055224  040504  040524  043440
13091  055232  053111  051505  053440
13092  055240  040510  020124  044123
13093  055246  052517  042114  041040
13094  055254  020105  044124  051105
13095  055262  006505     012
```

```
13096  055265     122  041505  044505          .ASCIZ  /RECEIVED DATA GIVES REGISTER CONTENTS AFTER COMMAND/
13097  055272  042526  020104  040504
13098  055300  040524  043440  053111
13099  055306  051505  051040  043505
13100  055314  051511  042524  020122
13101  055322  047503  052116  047105
13102  055330  051524  040440  052106
13103  055336  051105  041440  046517
13104  055344  040515  042116     000
13105  055351     127  044522  042524  EM36:   .ASCIZ  /WRITE DATA COMMAND CHANGED WRITE FROM BUFFER/
13106  055356  042040  052101  020101
13107  055364  047503  046515  047101
13108  055372  020104  044103  047101
13109  055400  042507  020104  051127
13110  055406  052111  020105  051106
13111  055414  046517  041040  043125
13112  055422  042506  000122
13113  055426  042523  045505  041440  EM37:   .ASCII  /SEEK COMMAND CAUSED IMPROPER REGISTER CHANGE/<15><12>
13114  055434  046517  040515  042116
13115  055442  041440  052501  042523
13116  055450  020104  046511  051120
13117  055456  050117  051105  051040
13118  055464  043505  051511  042524
13119  055472  020122  044103  047101
13120  055500  042507  005015
13121  055504  047507  042117  042040          .ASCII  /GOOD DATA GIVES WHAT SHOULD BE THERE/<15><12>
13122  055512  052101  020101  044507
13123  055520  042526  020123  044127
13124  055526  052101  051440  047510
13125  055534  046125  020104  042502
13126  055542  052040  042510  042522
13127  055550  005015
13128  055552  042522  042503  053111          .ASCIZ  /RECEIVED DATA GIVES REGISTER CONTENTS AFTER SEEK COMMAND/
13129  055560  042105  042040  052101
13130  055566  020101  044507  042526
13131  055574  020123  042522  044507
13132  055602  052123  051105  041440
13133  055610  047117  042524  052116
13134  055616  020123  043101  042524
13135  055624  020122  042523  045505
13136  055632  041440  046517  040515
13137  055640  042116     000
13138
13139  055643     127  044522  042524  EM40:   .ASCII  /WRITE CHECK CAUSED IMPROPER REGISTER CHANGE/<15><12>
13140  055650  041440  042510  045503
13141  055656  041440  052501  042523
13142  055664  020104  046511  051120
13143  055672  050117  051105  051040
13144  055700  043505  051511  042524
13145  055706  020122  044103  047101
13146  055714  042507  005015
13147  055720  047507  042117  042040          .ASCII  /GOOD DATA GIVES WHAT SHOULD BE THERE/<15><12>
13148  055726  052101  020101  044507
13149  055734  042526  020123  044127
13150  055742  052101  051440  047510
13151  055750  046125  020104  042502
```

```
13152  055756  052040  042510  042522
13153  055764  005015
13154  055766  042522  042503  053111          .ASCIZ  /RECEIVED DATA GIVES REGISTER CONTENTS AFTER COMMAND/
13155  055774  042105  042040  052101
13156  056002  020101  044507  042526
13157  056010  020123  042522  044507
13158  056016  052123  051105  041440
13159  056024  047117  042524  052116
13160  056032  020123  043101  042524
13161  056040  020122  047503  046515
13162  056046  047101  000104
13163  056052  047514  045503  047111  EM41:   .ASCII  /LOCKING OUT WRITE BY WRITE LOCK BUTTON CAUSED IMPROPER REGISTER CHANGE/
13164  056060  020107  052517  020124
13165  056066  051127  052111  020105
13166  056074  054502  053440  044522
13167  056102  042524  046040  041517
13168  056110  020113  052502  052124
13169  056116  047117  041440  052501
13170  056124  042523  020104  046511
13171  056132  051120  050117  051105
13172  056140  051040  043505  051511
13173  056146  042524  020122  044103
13174  056154  047101  042507  005015
13175  056162  047507  042117  042040          .ASCII  /GOOD DATA GIVES WHAT SHOULD BE THERE/<15><12>
13176  056170  052101  020101  044507
13177  056176  042526  020123  044127
13178  056204  052101  051440  047510
13179  056212  046125  020104  042502
13180  056220  052040  042510  042522
13181  056226  005015
13182  056230  042522  042503  053111          .ASCIZ  /RECEIVED DATA GIVES REGISTER CONTENTS AFTER WRITES WERE LOCKED OUT/
13183  056236  042105  042040  052101
13184  056244  020101  044507  042526
13185  056252  020123  042522  044507
13186  056260  052123  051105  041440
13187  056266  047117  042524  052116
13188  056274  020123  043101  042524
13189  056302  020122  051127  052111
13190  056310  051505  053440  051105
13191  056316  020105  047514  045503
13192  056324  042105  047440  052125
13193  056332     000
13194  056333     101  052124  046505  EM42:   .ASCII  /ATTEMPTING TO WRITE WITH WRITES LOCKED OUT CAUSED IMPROPER REGISTER CHA
13195  056340  052120  047111  020107
13196  056346  047524  053440  044522
13197  056354  042524  053440  052111
13198  056362  020110  051127  052111
13199  056370  051505  046040  041517
13200  056376  042513  020104  052517
13201  056404  020124  040503  051525
13202  056412  042105  044440  050115
13203  056420  047522  042520  020122
13204  056426  042522  044507  052123
13205  056434  051105  041440  040510
13206  056442  043516  006505     012
13207  056447     107  047517  020104          .ASCII  /GOOD DATA GIVES WHAT SHOULD BE THERE/<15><12>
```

```
13208  056454  040504  040524  043440
13209  056462  053111  051505  053440
13210  056470  040510  020124  044123
13211  056476  052517  042114  041040
13212  056504  020105  044124  051105
13213  056512  006505     012
13214  056515     122  041505  044505          .ASCIZ  /RECEIVED DATA GIVES REGISTER CONTENTS AFTER ATTEMPTED WRITE/
13215  056522  042526  020104  040504
13216  056530  040524  043440  053111
13217  056536  051505  051040  043505
13218  056544  051511  042524  020122
13219  056552  047503  052116  047105
13220  056560  051524  040440  052106
13221  056566  051105  040440  052124
13222  056574  046505  052120  042105
13223  056602  053440  044522  042524
13224  056610     000
13225  056611     127  044522  044524  EM43:   .ASCII  /WRITING WITH WRITES LOCKED OUT CHANGED DISK DATA/<15><12>
13226  056616  043516  053440  052111
13227  056624  020110  051127  052111
13228  056632  051505  046040  041517
13229  056640  042513  020104  052517
13230  056646  020124  044103  047101
13231  056654  042507  020104  044504
13232  056662  045523  042040  052101
13233  056670  006501     012
13234  056673     107  047517  020104          .ASCII  /GOOD DATA GIVES WHAT WAS ON DISK BEFORE WRITE (WITH WRITE LOCKED OUT)/<
13235  056700  040504  040524  043440
13236  056706  053111  051505  053440
13237  056714  040510  020124  040527
13238  056722  020123  047117  042040
13239  056730  051511  020113  042502
13240  056736  047506  042522  053440
13241  056744  044522  042524  024040
13242  056752  044527  044124  053440
13243  056760  044522  042524  046040
13244  056766  041517  042513  020104
13245  056774  052517  024524  005015
13246  057002  040527  020123  052101          .ASCII  /WAS ATTEMPTED/<15><12>
13247  057010  042524  050115  042524
13248  057016  006504     012
13249  057021     122  041505  044505          .ASCII  /RECEIVED DATA GIVES WHAT WAS READ BACK AFTER WRITE/<15><12>
13250  057026  042526  020104  040504
13251  057034  040524  043440  053111
13252  057042  051505  053440  040510
13253  057050  020124  040527  020123
13254  057056  042522  042101  041040
13255  057064  041501  020113  043101
13256  057072  042524  020122  051127
13257  057100  052111  006505     012
13258  057105     050  044527  044124          .ASCIZ  /(WITH WRITE LOCKED OUT) WAS ATTEMPTED/
13259  057112  053440  044522  042524
13260  057120  046040  041517  042513
13261  057126  020104  052517  024524
13262  057134  053440  051501  040440
13263  057142  052124  046505  052120
```

```
13264  057150  042105     000
13265  057153     105  040516  046102  EM44:   .ASCII  /ENABLING WRITES BY WRITE LOCK BUTTON CAUSED IMPROPER REGISTER CHANGE/<1
13266  057160  047111  020107  051127
13267  057166  052111  051505  041040
13268  057174  020131  051127  052111
13269  057202  020105  047514  045503
13270  057210  041040  052125  047524
13271  057216  020116  040503  051525
13272  057224  042105  044440  050115
13273  057232  047522  042520  020122
13274  057240  042522  044507  052123
13275  057246  051105  041440  040510
13276  057254  043516  006505     012
13277  057261     107  047517  020104          .ASCII  /GOOD DATA GIVES WHAT SHOULD BE THERE/<15><12>
13278  057266  040504  040524  043440
13279  057274  053111  051505  053440
13280  057302  040510  020124  044123
13281  057310  052517  042114  041040
13282  057316  020105  044124  051105
13283  057324  006505     012
13284  057327     122  041505  044505          .ASCII  /RECEIVED DATA GIVES REGISTER CONTENTS AFTER WRITE LOCK BUTTON/<15><12>
13285  057334  042526  020104  040504
13286  057342  040524  043440  053111
13287  057350  051505  051040  043505
13288  057356  051511  042524  020122
13289  057364  047503  052116  047105
13290  057372  051524  040440  052106
13291  057400  051105  053440  044522
13292  057406  042524  046040  041517
13293  057414  020113  052502  052124
13294  057422  047117  005015
13295  057426  047105  041101  042514          .ASCIZ  /ENABLED WRITES/
13296  057434  020104  051127  052111
13297  057442  051505     000
13298  057445     124  040522  051516  EM45:   .ASCII  /TRANSFERRING ON LAST BLOCK - CYLINDER 410, SECTOR 21, TRACK 18/<15><12>
13299  057452  042506  051122  047111
13300  057460  020107  047117  046040
13301  057466  051501  020124  046102
13302  057474  041517  020113  020055
13303  057502  054503  044514  042116
13304  057510  051105  032040  030061
13305  057516  020054  042523  052103
13306  057524  051117  031040  026061
13307  057532  052040  040522  045503
13308  057540  030440  006470     012
13309  057545     103  052501  042523          .ASCII  /CAUSED IMPROPER REGISTER CHANGE/<15><12>
13310  057552  020104  046511  051120
13311  057560  050117  051105  051040
13312  057566  043505  051511  042524
13313  057574  020122  044103  047101
13314  057602  042507  005015
13315  057606  047507  042117  042040          .ASCII  /GOOD DATA GIVES WHAT SHOULD BE THERE/<15><12>
13316  057614  052101  020101  044507
13317  057622  042526  020123  044127
13318  057630  052101  051440  047510
13319  057636  046125  020104  042502
```

```
13320  057644  052040  042510  042522
13321  057652  005015
13322  057654  042522  042503  053111          .ASCIZ  /RECEIVED DATA GIVES REGISTER CONTENTS AFTER TRANSFER/
13323  057662  042105  042040  052101
13324  057670  020101  044507  042526
13325  057676  020123  042522  044507
13326  057704  052123  051105  041440
13327  057712  047117  042524  052116
13328  057720  020123  043101  042524
13329  057726  020122  051124  047101
13330  057734  043123  051105     000
13331  057741     104  052101  020101  EM46:   .ASCII  /DATA READ FROM LAST BLOCK - CYLINDER 410, SECTOR 21, TRACK 18/<15><12>
13332  057746  042522  042101  043040
13333  057754  047522  020115  040514
13334  057762  052123  041040  047514
13335  057770  045503  026440  041440
13336  057776  046131  047111  042504
13337  060004  020122  030464  026060
13338  060012  051440  041505  047524
13339  060020  020122  030462  020054
13340  060026  051124  041501  020113
13341  060034  034061  005015
13342  060040  051511  044440  020116          .ASCIZ  /IS IN ERROR/
13343  060046  051105  047522  000122
13344  060054  051124  047101  043123  EM47:   .ASCII  /TRANSFERRING DATA FROM NONEXISTANT SECTOR CAUSED IMPROPER /<15><12>
13345  060062  051105  044522  043516
13346  060070  042040  052101  020101
13347  060076  051106  046517  047040
13348  060104  047117  054105  051511
13349  060112  040524  052116  051440
13350  060120  041505  047524  020122
13351  060126  040503  051525  042105
13352  060134  044440  050115  047522
13353  060142  042520  020122  005015
13354  060150  042522  044507  052123          .ASCII  /REGISTER CHANGE, GOOD DATA GIVES WHAT SHOULD BE THERE/<15><12>
13355  060156  051105  041440  040510
13356  060164  043516  026105  043440
13357  060172  047517  020104  040504
13358  060200  040524  043440  053111
13359  060206  051505  053440  040510
13360  060214  020124  044123  052517
13361  060222  042114  041040  020105
13362  060230  04412.  051105  006505
13363  060236     012
13364  060237     122  041505  044505          .ASCIZ  /RECEIVED DATA GIVES REGISTER CONTENTS AFTER ATTEMPTED TRANSFER/
13365  060244  042526  020104  040504
13366  060252  040524  043440  053111
13367  060260  051505  051040  043505
13368  060266  051511  042524  020122
13369  060274  047503  052116  047105
13370  060302  051524  040440  052106
13371  060310  051105  040440  052124
13372  060316  046505  052120  042105
13373  060324  052040  040522  051516
13374  060332  042506  000122
13375
```

```
13376  060336  051124  047101  043123  EM50:   .ASCII  /TRANSFERRING FROM NONEXISTANT SECTOR CAUSED DATA ERROR/<15><12>
13377  060344  051105  044522  043516
13378  060352  043040  047522  020115
13379  060360  047516  042516  044530
13380  060366  052123  047101  020124
13381  060374  042523  052103  051117
13382  060402  041440  052501  042523
13383  060410  020104  040504  040524
13384  060416  042440  051122  051117
13385  060424  005015
13386  060426  047507  042117  042040          .ASCII  /GOOD DATA GIVES WHAT SHOULD Be THERE/<15><12>
13387  060434  052101  020101  044507
13388  060442  042526  020123  044127
13389  060450  052101  051440  047510
13390  060456  046125  020104  042502
13391  060464  052040  042510  042522
13392  060472  005015
13393  060474  040502  020104  040504          .ASCIZ  /BAD DATA GIVES WHAT WAS IN BUFFER AFTER TRANSFER/
13394  060502  040524  043440  053111
13395  060510  051505  053440  040510
13396  060516  020124  040527  020123
13397  060524  047111  041040  043125
13398  060532  042506  020122  043101
13399  060540  042524  020122  051124
13400  060546  047101  043123  051105
13401  060554     000
13402  060555     107  053111  047111  EM51:   .ASCII  /GIVING ILLEGAL FUNCTION CAUSED IMPROPER REGISTER CHANGE/<15><12>
13403  060562  020107  046111  042514
13404  060570  040507  020114  052506
13405  060576  041516  044524  047117
13406  060604  041440  052501  042523
13407  060612  020104  046511  051120
13408  060620  050117  051105  051040
13409  060626  043505  051511  042524
13410  060634  020122  044103  047101
13411  060642  042507  005015
13412  060646  047507  042117  042040          .ASCII  /GOOD DATA GIVES WHAT SHOULD BE THERE/<15><12>
13413  060654  052101  020101  044507
13414  060662  042526  020123  044127
13415  060670  052101  051440  047510
13416  060676  046125  020104  042502
13417  060704  052040  042510  042522
13418  060712  005015
13419  060714  042522  042503  053111          .ASCIZ  /RECEIVED DATA GIVES REGISTER CONTENTS AFTER ILLEGAL FUNCTION IS GIVEN/
13420  060722  042105  042040  052101
13421  060730  020101  044507  042526
13422  060736  020123  042522  044507
13423  060744  052123  051105  041440
13424  060752  047117  042524  052116
13425  060760  020123  043101  042524
13426  060766  020122  046111  042514
13427  060774  040507  020114  052506
13428  061002  041516  044524  047117
13429  061010  044440  020123  044507
13430  061016  042526  000116
13431  061022  051127  052111  020105  EM52:   .ASCII  /WRITE DATA COMMAND ON NONEXISTANT SECTOR CAUSED IMPROPER REG. CHANGE/<1
```

```
 13432  061030  040504  040524  041440
 13433  061036  046517  040515  042116
 13434  061044  047440  020116  047516
 13435  061052  042516  044530  052123
 13436  061060  047101  020124  042523
 13437  061066  052103  051117  041440
 13438  061074  052501  042523  020104
 13439  061102  046511  051120  050117
 13440  061110  051105  051040  043505
 13441  061116  020056  044103  047101
 13442  061124  042507  005015
 13443  061130  047507  042117  042040          .ASCII  /GOOD DATA GIVES WHAT SHOULD BE THERE/<15><12>
 13444  061136  052101  020101  044507
 13445  061144  042526  020123  044127
 13446  061152  052101  051440  047510
 13447  061160  046125  020104  042502
 13448  061166  052040  042510  042522
 13449  061174  005015
 13450  061176  042522  042503  053111          .ASCIZ  /RECEIVED DATA GIVES REGISTER CONTENTS AFTER ATTEMPTED WRITE DATA/
 13451  061204  042105  042040  052101
 13452  061212  020101  044507  042526
 13453  061220  020123  042522  044507
 13454  061226  052123  051105  041440
 13455  061234  047117  042524  052116
 13456  061242  020123  043101  042524
 13457  061250  020122  052101  042524
 13458  061256  050115  042524  020104
 13459  061264  051127  052111  020105
 13460  061272  040504  040524     000
 13461  061277     122  040505  020104  EM53:   .ASCIZ  /READ HEADER AND DATA AFTER A SEARCH CAUSED DATA ERROR/
 13462  061304  042510  042101  051105
 13463  061312  040440  042116  042040
 13464  061320  052101  020101  043101
 13465  061326  042524  020122  020101
 13466  061334  042523  051101  044103
 13467  061342  041440  052501  042523
 13468  061350  020104  040504  040524
 13469  061356  042440  051122  051117
 13470  061364     000
 13471  061365     101  052124  046505  EM54:   .ASCII  /ATTEMPTING COMMAND WITH INVALID ADDRESS CAUSED IMPROPER REGISTER CHANGE
 13472  061372  052120  047111  020107
 13473  061400  047503  046515  047101
 13474  061406  020104  044527  044124
 13475  061414  044440  053116  046101
 13476  061422  042111  040440  042104
 13477  061430  042522  051523  041440
 13478  061436  052501  042523  020104
 13479  061444  046511  051120  050117
 13480  061452  051105  051040  043505
 13481  061460  051511  042524  020122
 13482  061466  044103  047101  042507
 13483  061474  005015
 13484  061476  047507  042117  042040          .ASCII  /GOOD DATA GIVES WHAT SHOULD BE THERE/<15><12>
 13485  061504  052101  020101  044507
 13486  061512  042526  020123  044127
 13487  061520  052101  051440  047510
```

```
13488  061526  046125  020104  042502
13489  061534  052040  042510  042522
13490  061542  005015
13491  061544  042522  042503  053111          .ASCIZ  /RECEIVED DATA GIVES REGISTER CONTENTS AFTER OPERATION/
13492  061552  042105  042040  052101
13493  061560  020101  044507  042526
13494  061566  020123  042522  044507
13495  061574  052123  051105  041440
13496  061602  047117  042524  052116
13497  061610  020123  043101  042524
13498  061616  020122  050117  051105
13499  061624  052101  047511  000116
13500  061632  051127  052111  047111  EM55:   .ASCII  /WRITING OR READING WITH EXPECTED ADDRESS OVERFLOW ERROR/<15><12>
13501  061640  020107  051117  051040
13502  061646  040505  044504  043516
13503  061654  053440  052111  020110
13504  061662  054105  042520  052103
13505  061670  042105  040440  042104
13506  061676  042522  051523  047440
13507  061704  042526  043122  047514
13508  061712  020127  051105  047522
13509  061720  006522     012
13510  061723     103  052501  042523          .ASCII  /CAUSED IMPROPER REGISTER CHANGE/<15><12>
13511  061730  020104  046511  051120
13512  061736  050117  051105  051040
13513  061744  043505  051511  042524
13514  061752  020122  044103  047101
13515  061760  042507  005015
13516  061764  047507  042117  042040          .ASCII  /GOOD DATA GIVES WHAT SHOULD BE THERE/<15><12>
13517  061772  052101  020101  044507
13518  062000  042526  020123  044127
13519  062006  052101  051440  047510
13520  062014  046125  020104  042502
13521  062022  052040  042510  042522
13522  062030  005015
13523  062032  042522  042503  053111          .ASCIZ  /RECEIVED DATA GIVES REGISTER CONTENTS AFTER OPERATION/
13524  062040  042105  042040  052101
13525  062046  020101  044507  042526
13526  062054  020123  042522  044507
13527  062062  052123  051105  041440
13528  062070  047117  042524  052116
13529  062076  020123  043101  042524
13530  062104  020122  050117  051105
13531  062112  052101  047511  000116
13532  062120  040504  040524  051040  EM56:   .ASCII  /DATA READ WITH AN EXPECTED ADDRESS OVERFLOW ERROR IS INCORRECT/<15><12>
13533  062126  040505  020104  044527
13534  062134  044124  040440  020116
13535  062142  054105  042520  052103
13536  062150  042105  040440  042104
13537  062156  042522  051523  047440
13538  062164  042526  043122  047514
13539  062172  020127  051105  047522
13540  062200  020122  051511  044440
13541  062206  041516  051117  042522
13542  062214  052103  005015
13543  062220  047527  042122  047040          .ASCII  /WORD NO. 1 TO 260 SHOULD BE READ, WORD NO 261 TO 266 SHOULD/<15><12>
```

```
13544  062226  027117  030440  052040
13545  062234  020117  033062  020060
13546  062242  044123  052517  042114
13547  062250  041040  020105  042522
13548  062256  042101  020054  047527
13549  062264  042122  047040  020117
13550  062272  033062  020061  047524
13551  062300  031040  033066  051440
13552  062306  047510  046125  006504
13553  062314     012
13554  062315     102  020105  044103          .ASCIZ  /BE CHANGED/
13555  062322  047101  042507  000104
13556  062330  052101  042524  050115  EM57:  .ASCII  /ATTEMPTING DATA COMMAND WITH WRONG FORMAT BIT CAUSED/<15><12>
13557  062336  044524  043516  042040
13558  062344  052101  020101  047503
13559  062352  046515  047101  020104
13560  062360  044527  044124  053440
13561  062366  047522  043516  043040
13562  062374  051117  040515  020124
13563  062402  044502  020124  040503
13564  062410  051525  042105  005015
13565  062416  046511  051120  050117          .ASCII  /IMPROPER REGISTER CHANGE/<15><12>
13566  062424  051105  051040  043505
13567  062432  051511  042524  020122
13568  062440  044103  047101  042507
13569  062446  005015
13570  062450  047507  042117  042040          .ASCII  /GOOD DATA GIVES WHAT SHOULD BE THERE/<15><12>
13571  062456  052101  020101  044507
13572  062464  042526  020123  044127
13573  062472  052101  051440  047510
13574  062500  046125  020104  042502
13575  062506  052040  042510  042522
13576  062514  005015
13577  062516  042522  042503  053111          .ASCIZ  /RECEIVED DATA GIVES REGISTER CONTENTS AFTER ATTEMPTED DATA TRANSFER/
13578  062524  042105  042040  052101
13579  062532  020101  044507  042526
13580  062540  020123  042522  044507
13581  062546  052123  051105  041440
13582  062554  047117  042524  052116
13583  062562  020123  043101  042524
13584  062570  020122  052101  042524
13585  062576  050115  042524  020104
13586  062604  040504  040524  052040
13587  062612  040522  051516  042506
13588  062620  000122
13589
13590  062622  052101  042524  050115  EM60:  .ASCII  /ATTEMPTING TO MODIFY REGISTER DURING AN OPERATION CAUSED IMPROPER/<15><
13591  062630  044524  043516  052040
13592  062636  020117  047515  044504
13593  062644  054506  051040  043505
13594  062652  051511  042524  020122
13595  062660  052504  044522  043516
13596  062666  040440  020116  050117
13597  062674  051105  052101  047511
13598  062702  020116  040503  051525
13599  062710  042105  044440  050115
```

```
 13600  062716  047522  042520  006522
 13601  062724     012
 13602  062725     122  043505  051511          .ASCII  /REGISTER CHANGE. GOOD DATA GIVES WHAT SHOULD BE THERE/<15><12>
 13603  062732  042524  020122  044103
 13604  062740  047101  042507  020056
 13605  062746  047507  042117  042040
 13606  062754  052101  020101  044507
 13607  062762  042526  020123  044127
 13608  062770  052101  051440  047510
 13609  062776  046125  020104  042502
 13610  063004  052040  042510  042522
 13611  063012  005015
 13612  063014  042522  042503  053111          .ASCII  /RECEIVED DATA GIVES REGISTER CONTENTS AFTER OPERATION WAS ATTEMPTED/<15
 13613  063022  042105  042040  052101
 13614  063030  020101  044507  042526
 13615  063036  020123  042522  044507
 13616  063044  052123  051105  041440
 13617  063052  047117  042524  052116
 13618  063060  020123  043101  042524
 13619  063066  020122  050117  051105
 13620  063074  052101  047511  020116
 13621  063102  040527  020123  052101
 13622  063110  042524  050115  042524
 13623  063116  006504     012
 13624  063121     115  042117  043111          .ASCIZ  /MODIFYING REG GIVES ADDRESS OF REGISTER BEING MODIFIED WHICH CAUSED ERR
 13625  063126  044531  043516  051040
 13626  063134  043505  043440  053111
 13627  063142  051505  040440  042104
 13628  063150  042522  051523  047440
 13629  063156  020106  042522  044507
 13630  063164  052123  051105  041040
 13631  063172  044505  043516  046440
 13632  063200  042117  043111  042511
 13633  063206  020104  044127  041511
 13634  063214  020110  040503  051525
 13635  063222  042105  042440  051122
 13636  063230  051117     000
 13637  063233     122  041510  030523  EM61:   .ASCIZ  /RHCS1 HAS SOME INCORRECT STATUS BITS = 1, OR = 0/
 13638  063240  044040  051501  051440
 13639  063246  046517  020105  047111
 13640  063254  047503  051122  041505
 13641  063262  020124  052123  052101
 13642  063270  051525  041040  052111
 13643  063276  020123  020075  026061
 13644  063304  047440  020122  020075
 13645  063312  000060
 13646  063314  044122  051504  020061  EM62:   .ASCIZ  /RHDS1 HAS SOME INCORRECT STATUS BITS = 1, OR = 0/
 13647  063322  040510  020123  047523
 13648  063330  042515  044440  041516
 13649  063336  051117  042522  052103
 13650  063344  051440  040524  052524
 13651  063352  020123  044502  051524
 13652  063360  036440  030440  020054
 13653  063366  051117  036440  030040
 13654  063374     000
 13655  063375     122  042110  030523  EM63:   .ASCIZ  /RHDS1 CONTENTS DURING COMMAND WERE IN ERROR/
```

```
13656  063402  041440  047117  042524
13657  063410  052116  020123  052504
13658  063416  044522  043516  041440
13659  063424  046517  040515  042116
13660  063432  053440  051105  020105
13661  063440  047111  042440  051122
13662  063446  051117     000
13663  063451     122  041505  046101  EM64:   .ASCII  /RECALIBRATE COMMAND CAUSED IMPROPER REGISTER CHANGE/<15><12>
13664  063456  041111  040522  042524
13665  063464  041440  046517  040515
13666  063472  042116  041440  052501
13667  063500  042523  020104  046511
13668  063506  051120  050117  051105
13669  063514  051040  043505  051511
13670  063522  042524  020122  044103
13671  063530  047101  042507  005015
13672  063536  047507  042117  042040          .ASCII  /GOOD DATA GIVES WHAT SHOULD BE THERE/<15><12>
13673  063544  052101  020101  044507
13674  063552  042526  020123  044127
13675  063560  052101  051440  047510
13676  063566  046125  020104  042502
13677  063574  052040  042510  042522
13678  063602  005015
13679  063604  042522  042503  053111          .ASCIZ  /RECEIVED DATA GIVES REGISTER CONTENTS AFTER COMMAND/
13680  063612  042105  042040  052101
13681  063620  020101  044507  042526
13682  063626  020123  042522  044507
13683  063634  052123  051105  041440
13684  063642  047117  042524  052116
13685  063650  020123  043101  042524
13686  063656  020122  047503  046515
13687  063664  047101  000104
13688  063670  047111  042524  051122  EM65:   .ASCIZ  /INTERRUPT FAILING/
13689  063676  050125  020124  040506
13690  063704  046111  047111  000107
13691  063712  042510  042101  051105  EM66:   .ASCII  /HEADER AND DATA COMMAND FOR HEAD SELECTION TEST CAUSED/<15><12>
13692  063720  040440  042116  042040
13693  063726  052101  020101  047503
13694  063734  046515  047101  020104
13695  063742  047506  020122  042510
13696  063750  042101  051440  046105
13697  063756  041505  044524  047117
13698  063764  052040  051505  020124
13699  063772  040503  051525  042105
13700  064000  005015
13701  064002  051105  047522  020122          .ASCII  /ERROR - RHDST GIVES TRACK BEING WRITTEN OR READ ON CYL 0, SCTR 0/<15><1
13702  064010  020055  044122  051504
13703  064016  020124  044507  042526
13704  064024  020123  051124  041501
13705  064032  020113  042502  047111
13706  064040  020107  051127  052111
13707  064046  042524  020116  051117
13708  064054  051040  040505  020104
13709  064062  047117  041440  046131
13710  064070  030040  020054  041523
13711  064076  051124  030040  005015
```

```
13712  064104  042522  042101  044040  EM67:   .ASCII  /READ HEADER AND DATA ERROR IN HEAD SELECTION TEST./<12><15>
13713  064112  040505  042504  020122
13714  064120  047101  020104  040504
13715  064126  040524  042440  051122
13716  064134  051117  044440  020116
13717  064142  042510  042101  051440
13718  064150  046105  041505  044524
13719  064156  047117  052040  051505
13720  064164  027124  006412
13721  064170  044506  051522  020124          .ASCII  /FIRST FOUR WORD NUMBERS ARE THE HEADER./<12><15>
13722  064176  047506  051125  053440
13723  064204  051117  020104  052516
13724  064212  041115  051105  020123
13725  064220  051101  020105  044124
13726  064226  020105  042510  042101
13727  064234  051105  005056     015
13728  064241     127  051117  020104          .ASCII  /WORD NUMBERS 5 TO 260 ARE DATA WORDS,/<12><15>
13729  064246  052516  041115  051105
13730  064254  020123  020065  047524
13731  064262  031040  030066  040440
13732  064270  042522  042040  052101
13733  064276  020101  047527  042122
13734  064304  026123  006412
13735  064310  047101  020104  047111          .ASCIZ  /AND IN DATA WORDS BITS 4,5,6,7,8 GIVE TRACK NUMBER./
13736  064316  042040  052101  020101
13737  064324  047527  042122  020123
13738  064332  044502  051524  032040
13739  064340  032454  033054  033454
13740  064346  034054  043440  053111
13741  064354  020105  051124  041501
13742  064362  020113  052516  041115
13743  064370  051105  000056
13744
13745  064374  042522  042101  044040  EM70:   .ASCII  /READ HEADER AND DATA ERROR IN/<15><12>
13746  064402  040505  042504  020122
13747  064410  047101  020104  040504
13748  064416  040524  042440  051122
13749  064424  051117  044440  006516
13750  064432     012
13751  064433     104  043111  042506          .ASCII  /DIFFERENCE LINE TEST/<15><12>
13752  064440  042522  041516  020105
13753  064446  044514  042516  052040
13754  064454  051505  006524     012
13755  064461     127  051117  020104          .ASCII  /WORD NOS 1-4 GIVE HEADER/<15><12>
13756  064466  047516  020123  026461
13757  064474  020064  044507  042526
13758  064502  044040  040505  042504
13759  064510  006522     012
13760  064513     127  051117  020104          .ASCIZ  /WORD NOS 5-260 GIVE DATA WHICH IS THE CYLINDER ADDRESS/
13761  064520  047516  020123  026465
13762  064526  033062  020060  044507
13763  064534  042526  042040  052101
13764  064542  020101  044127  041511
13765  064550  020110  051511  052040
13766  064556  042510  041440  046131
13767  064564  047111  042504  020122
```

```
13768  064572  042101  051104  051505
13769  064600  000123
13770  064602  047506  041522  047111  EM71:   .ASCII  /FORCING OPI BY 3 INDEX PULSES/<15><12>
13771  064610  020107  050117  020111
13772  064616  054502  031440  044440
13773  064624  042116  054105  050040
13774  064632  046125  042523  006523
13775  064640     012
13776  064641     103  052501  042523          .ASCII  /CAUSED IMPROPER REGISTER CHANGE/<15><12>
13777  064646  020104  046511  051120
13778  064654  050117  051105  051040
13779  064662  043505  051511  042524
13780  064670  020122  044103  047101
13781  064676  042507  005015
13782  064702  047507  042117  042040          .ASCII  /GOOD DATA GIVES WHAT SHOULD BE THERE/<15><12>
13783  064710  052101  020101  044507
13784  064716  042526  020123  044127
13785  064724  052101  051440  047510
13786  064732  046125  020104  042502
13787  064740  052040  042510  042522
13788  064746  005015
13789  064750  042522  042503  053111          .ASCIZ  /RECEIVED DATA GIVES REGISTER CONTENTS AFTER 3 INDEX PULSES/
13790  064756  042105  042040  052101
13791  064764  020101  044507  042526
13792  064772  020123  042522  044507
13793  065000  052123  051105  041440
13794  065006  047117  042524  052116
13795  065014  020123  043101  042524
13796  065022  020122  020063  047111
13797  065030  042504  020130  052520
13798  065036  051514  051505     000
13799  065043     127  044510  042514  EM72:   .ASCII  /WHILE USING UNIBUS B/<15><12>
13800  065050  052440  044523  043516
13801  065056  052440  044516  052502
13802  065064  020123  006502     012
13803  065071     122  040505  020104          .ASCII  /READ DATA CAUSED IMPROPER REGISTER CHANGE/<15><12>
13804  065076  040504  040524  041440
13805  065104  052501  042523  020104
13806  065112  046511  051120  050117
13807  065120  051105  051040  043505
13808  065126  051511  042524  020122
13809  065134  044103  047101  042507
13810  065142  005015
13811  065144  047507  042117  042040          .ASCII  /GOOD DATA GIVES WHAT SHOULD BE THERE/<15><12>
13812  065152  052101  020101  044507
13813  065160  042526  020123  044127
13814  065166  052101  051440  047510
13815  065174  046125  020104  042502
13816  065202  052040  042510  042522
13817  065210  005015
13818  065212  042522  042503  053111          .ASCIZ  /RECEIVED DATA GIVES WHAT WAS THERE AFTER COMMAND/
13819  065220  042105  042040  052101
13820  065226  020101  044507  042526
13821  065234  020123  044127  052101
13822  065242  053440  051501  052040
13823  065250  042510  042522  040440
```

```
13824  065256  052106  051105  041440
13825  065264  046517  040515  042116
13826  065272     000
13827  065273     127  044510  042514  EM73:   .ASCII  /WHILE USING UNIBUS B/<15><12>
13828  065300  052440  044523  043516
13829  065306  052440  044516  052502
13830  065314  020123  006502     012
13831  065321     122  040505  020104          .ASCIZ  /READ DATA INCORRECT/
13832  065326  040504  040524  044440
13833  065334  041516  051117  042522
13834  065342  052103     000
13835  065345     127  044510  042514  EM74:   .ASCII  /WHILE USING UNIBUS B/<15><12>
13836  065352  052440  044523  043516
13837  065360  052440  044516  052502
13838  065366  020123  006502     012
13839  065373     127  044522  042524          .ASCII  /WRITE DATA COMMAND CAUSED IMPROPER REGISTER CHANGE/<15><12>
13840  065400  042040  052101  020101
13841  065406  047503  046515  047101
13842  065414  020104  040503  051525
13843  065422  042105  044440  050115
13844  065430  047522  042520  020122
13845  065436  042522  044507  052123
13846  065444  051105  041440  040510
13847  065452  043516  006505     012
13848  065457     107  047517  020104          .ASCII  /GOOD DATA GIVES WHAT SHOULD BE THERE/<15><12>
13849  065464  040504  040524  043440
13850  065472  053111  051505  053440
13851  065500  040510  020124  044123
13852  065506  052517  042114  041040
13853  065514  020105  044124  051105
13854  065522  006505     012
13855  065525     122  041505  044505          .ASCIZ  /RECEIVED DATA GIVES REGISTER CONTENTS AFTER COMMAND/
13856  065532  042526  020104  040504
13857  065540  040524  043440  053111
13858  065546  051505  051040  043505
13859  065554  051511  042524  020122
13860  065562  047503  052116  047105
13861  065570  051524  040440  052106
13862  065576  051105  041440  046517
13863  065604  040515  042116     000
13864  065611     127  044510  042514  EM75:   .ASCII  /WHILE USING UNIBUS B/<15><12>
13865  065616  052440  044523  043516
13866  065624  052440  044516  052502
13867  065632  020123  006502     012
13868  065637     127  044522  042524          .ASCIZ  /WRITE DATA COMMAND CHANGED WRITE FROM BUFFER/
13869  065644  042040  052101  020101
13870  065652  047503  046515  047101
13871  065660  020104  044103  047101
13872  065666  042507  020104  051127
13873  065674  052111  020105  051106
13874  065702  046517  041040  043125
13875  065710  042506  000122
13876  065714  044127  046111  020105  EM76:   .ASCII  /WHILE USING UNIBUS B/<15><12>
13877  065722  051525  047111  020107
13878  065730  047125  041111  051525
13879  065736  041040  005015
```

```
13880  065742  051127  052111  020105          .ASCII  /WRITE CHECK CAUSED IMPROPER REGISTER CHANGE/<15><12>
13881  065750  044103  041505  020113
13882  065756  040503  051525  042105
13883  065764  044440  050115  047522
13884  065772  042520  020122  042522
13885  066000  044507  052123  051105
13886  066006  041440  040510  043516
13887  066014  006505     012
13888  066017     107  047517  020104          .ASCII  /GOOD DATA GIVES WHAT SHOULD BE THERE/<15><12>
13889  066024  040504  040524  043440
13890  066032  053111  051505  053440
13891  066040  040510  020124  044123
13892  066046  052517  042114  041040
13893  066054  020105  044124  051105
13894  066062  006505     012
13895  066065     122  041505  044505          .ASCIZ  /RECEIVED DATA GIVES REGISTER CONTENTS AFTER COMMAND/
13896  066072  042526  020104  040504
13897  066100  040524  043440  053111
13898  066106  051505  051040  043505
13899  066114  051511  042524  020122
13900  066122  047503  052116  047105
13901  066130  051524  040440  052106
13902  066136  051105  041440  046517
13903  066144  040515  042116     000
13904  066151     120  042522  047514  EM77:   .ASCII  /PRELOADING 'RHCC' PRIOR TO DOING NEXT TEST DOES NOT PRODUCE CORRECT RES
13905  066156  042101  047111  020107
13906  066164  051047  041510  023503
13907  066172  050040  044522  051117
13908  066200  052040  020117  047504
13909  066206  047111  020107  042516
13910  066214  052130  052040  051505
13911  066222  020124  047504  051505
13912  066230  047040  052117  050040
13913  066236  047522  052504  042503
13914  066244  041440  051117  042522
13915  066252  052103  051040  051505
13916  066260  046125  006524     012
13917  066265     124  042510  042522          .ASCIZ  /THEREFORE NEXT TEST RESULTS ARE SUSPECT WITH REGARD TO 'RHCC' CONTENTS/
13918  066272  047506  042522  047040
13919  066300  054105  020124  042524
13920  066306  052123  051040  051505
13921  066314  046125  051524  040440
13922  066322  042522  051440  051525
13923  066330  042520  052103  053440
13924  066336  052111  020110  042522
13925  066344  040507  042122  052040
13926  066352  020117  051047  041510
13927  066360  023503  041440  047117
13928  066366  042524  052116  000123
13929
13930  066374  042101  051104  051505  EM100:  .ASCIZ  /ADDRESS PLUG CHANGE RESULTED IN BAD REGISTER DATA/
13931  066402  020123  046120  043525
13932  066410  041440  040510  043516
13933  066416  020105  042522  052523
13934  066424  052114  042105  044440
13935  066432  020116  040502  020104
```

```
13936  066440  042522  044507  052123
13937  066446  051105  042040  052101
13938  066454  000101
13939  066456  047125  052111  042040  EM101:  .ASCIZ  /UNIT DID NOT GO OFFLINE WHEN ADDRESS PLUG REMOVED/
13940  066464  042111  047040  052117
13941  066472  043440  020117  043117
13942  066500  046106  047111  020105
13943  066506  044127  047105  040440
13944  066514  042104  042522  051523
13945  066522  050040  052514  020107
13946  066530  042522  047515  042526
13947  066536  000104
13948  066540  047125  052111  047040  EM102:  .ASCIZ  /UNIT NOT AVAILABLE AFTER ADDRESS PLUG REPLACED/
13949  066546  052117  040440  040526
13950  066554  046111  041101  042514
13951  066562  040440  052106  051105
13952  066570  040440  042104  042522
13953  066576  051523  050040  052514
13954  066604  020107  042522  046120
13955  066612  041501  042105     000
13956  066617     122  043505  051511  EM103:  .ASCIZ  /REGISTER CONTENTS INCORRECT BEFORE A DIAG MODE SEEK/
13957  066624  042524  020122  047503
13958  066632  052116  047105  051524
13959  066640  044440  041516  051117
13960  066646  042522  052103  041040
13961  066654  043105  051117  020105
13962  066662  020101  044504  043501
13963  066670  046440  042117  020105
13964  066676  042523  045505     000
13965  066703     122  043505  051511  EM104:  .ASCIZ  /REGISTER CONTENTS INCORRECT AFTER A DIAG MODE SEEK/
13966  066710  042524  020122  047503
13967  066716  052116  047105  051524
13968  066724  044440  041516  051117
13969  066732  042522  052103  040440
13970  066740  052106  051105  040440
13971  066746  042040  040511  020107
13972  066754  047515  042504  051440
13973  066762  042505  000113
13974  066766  050040  047522  051107  NOUSE:  .ASCIZ  / PROGRAMMABLE-DRIVE WILL NOT BE USED/
13975  066774  046501  040515  046102
13976  067002  026505  051104  053111
13977  067010  020105  044527  046114
13978  067016  047040  052117  041040
13979  067024  020105  051525  042105
13980  067032     000
```

```
13981
13982
13983  067033   106  052101  046101  CPHALT: .ASCII  /FATAL ERROR - SEE DOCUMENT LISTING/<15><12><15><12>
13984  067040  042440  051122  051117
13985  067046  026440  051440  042505
13986  067054  042040  041517  046525
13987  067062  047105  020124  044514
13988  067070  052123  047111  006507
13989  067076  006412    012
13990  067101   124  042510  041440          .ASCII  /THE CONTROLLER OR DEVICE HAS GONE OFFLINE, LOST/<15><12>
13991  067106  047117  051124  046117
13992  067114  042514  020122  051117
13993  067122  042040  053105  041511
13994  067130  020105  040510  020123
13995  067136  047507  042516  047440
13996  067144  043106  044514  042516
13997  067152  020054  047514  052123
13998  067160  005015
13999  067162  051047  040505  054504          .ASCII  /'READY', BECOME UNAVAILABLE, OR HAS STATUS BITS/<15><12>
14000  067170  026047  041040  041505
14001  067176  046517  020105  047125
14002  067204  053101  044501  040514
14003  067212  046102  026105  047440
14004  067220  020122  040510  020123
14005  067226  052123  052101  051525
14006  067234  041040  052111  006523
14007  067242    012
14008  067243   127  044510  044103          .ASCIZ  /WHICH CANNOT BE CLEARED/
14009  067250  041440  047101  047516
14010  067256  020124  042502  041440
14011  067264  042514  051101  042105
14012  067272    000
14013
14014
14015
14016
14017  067273   120  020103  020040  DH1:    .ASCII  /PC      TEST    WAT     BIT     REG     REG     RHCS1/<15><12>
14018  067300  020040  052040  051505
14019  067306  020124  020040  053440
14020  067314  052101  020040  020040
14021  067322  041040  052111  020040
14022  067330  020040  051040  043505
14023  067336  020040  020040  051040
14024  067344  043505  020040  020040
14025  067352  051040  041510  030523
14026  067360  005015
14027  067362  020040  020040  020040          .ASCIZ  /       NO      PC      EXPECT  ADDRESS CONTENT CONTENT/
14028  067370  020040  047516  020040
14029  067376  020040  020040  041520
14030  067404  020040  020040  020040
14031  067412  054105  042520  052103
14032  067420  020040  042101  051104
14033  067426  051505  020123  047503
14034  067434  052116  047105  020124
14035  067442  047503  052116  047105
14036  067450  000124
```

```
14037  067452  041520  020040  020040  DH4:    .ASCII  /PC      TEST    WAT     BIT     REG     TIME IN/<15><12>
14038  067460  020040  042524  052123
14039  067466  020040  020040  040527
14040  067474  020124  020040  020040
14041  067502  044502  020124  020040
14042  067510  020040  042522  020107
14043  067516  020040  020040  044524
14044  067524  042515  044440  006516
14045  067532     012
14046  067533     040  020040  020040          .ASCIZ  /       NO      PC      EXPECT  ADDRESS 10 MSEC/
14047  067540  020040  047040  020117
14048  067546  020040  020040  050040
14049  067554  020103  020040  020040
14050  067562  042440  050130  041505
14051  067570  020124  040440  042104
14052  067576  042522  051523  030440
14053  067604  020060  051515  041505
14054  067612     000
14055  067613     120  020103  020040  DH5:    .ASCII  /PC      TEST    REG     GOOD    RECEIVED/<15><12>
14056  067620  020040  052040  051505
14057  067626  020124  020040  051040
14058  067634  043505  020040  020040
14059  067642  043440  047517  020104
14060  067650  020040  051040  041505
14061  067656  044505  042526  006504
14062  067664     012
14063  067665     040  020040  020040          .ASCIZ  /       NO      ADDRESS DATA    DATA/
14064  067672  020040  047040  020117
14065  067700  020040  020040  040440
14066  067706  042104  042522  051523
14067  067714  042040  052101  020101
14068  067722  020040  042040  052101
14069  067730  000101
14070  067732  041520  020040  020040  DH6:    .ASCII  /PC      TEST    REG     RECEIVED/<15><12>
14071  067740  020040  042524  052123
14072  067746  020040  020040  042522
14073  067754  020107  020040  020040
14074  067762  042522  042503  053111
14075  067770  042105  005015
14076  067774  020040  020040  020040          .ASCIZ  /       NO      ADDRESS DATA/
14077  070002  020040  047516  020040
14078  070010  020040  020040  042101
14079  070016  051104  051505  020123
14080  070024  040504  040524     000
14081  070031     120  020103  020040  DH7:    .ASCIZ  /PC      TEST    REG     ADDRESS/
14082  070036  020040  052040  051505
14083  070044  020124  020040  051040
14084  070052  043505  020040  020040
14085  070060  040440  042104  042522
14086  070066  051523     000
14087
14088  070071     120  020103  020040  DH10:   .ASCII  /PC      TEST    FAILING CONTENT CONTENT CONTENT CONTENT/<15><12>
14089  070076  020040  052040  051505
14090  070104  020124  020040  043040
14091  070112  044501  044514  043516
14092  070120  041440  047117  042524
```

```
14093  070126  052116  041440  047117
14094  070134  042524  052116  041440
14095  070142  047117  042524  052116
14096  070150  041440  047117  042524
14097  070156  052116  005015
14098  070162  020040  020040  020040          .ASCIZ  /       NO      REG.    RHCS1   RHCS1   RHDS1   RHER1/
14099  070170  020040  047516  020040
14100  070176  020040  020040  042522
14101  070204  027107  020040  020040
14102  070212  044122  051503  020061
14103  070220  020040  044122  051503
14104  070226  020061  020040  044122
14105  070234  051504  020061  020040
14106  070242  044122  051105  000061
14107  070250  041520  020040  020040  DH26:   .ASCII  /PC      TEST    CONT OF CONT OF CONT OF CONT OF CONT OF CONT OF/<15><12
14108  070256  020040  042524  052123
14109  070264  020040  020040  047503
14110  070272  052116  047440  020106
14111  070300  047503  052116  047440
14112  070306  020106  047503  052116
14113  070314  047440  020106  047503
14114  070322  052116  047440  020106
14115  070330  047503  052116  047440
14116  070336  020106  047503  052116
14117  070344  047440  006506     012
14118  070351     040  020040  020040          .ASCIZ  /       NO      RHCS1   RHCS2   RHDS1   RHER1   RHER2   RHER3/
14119  070356  020040  047040  020117
14120  070364  020040  020040  051040
14121  070372  041510  030523  020040
14122  070400  051040  041510  031123
14123  070406  020040  051040  042110
14124  070414  030523  020040  051040
14125  070422  042510  030522  020040
14126  070430  051040  042510  031122
14127  070436  020040  051040  042510
14128  070444  031522     000
14129  070447     120  020103  020040  DH30:   .ASCII  /PC      TEST    WORD    GOOD    BAD/<15><12>
14130  070454  020040  052040  051505
14131  070462  020124  020040  053440
14132  070470  051117  020104  020040
14133  070476  043440  047517  020104
14134  070504  020040  041040  042101
14135  070512  005015
14136  070514  020040  020040  020040          .ASCIZ  /       NO      NO      DATA    DATA/<15><12>
14137  070522  020040  047516  020040
14138  070530  020040  020040  047516
14139  070536  020040  020040  020040
14140  070544  040504  040524  020040
14141  070552  020040  040504  040524
14142  070560  005015     000
14143
14144  070563     120  020103  020040  DH51:   .ASCII  /PC      TEST    REG     GOOD    RECVD   ILLEGL/<15><12>
14145  070570  020040  052040  051505
14146  070576  020124  020040  051040
14147  070604  043505  020040  020040
14148  070612  043440  047517  020104
```

```
14149  070620  020040  051040  041505
14150  070626  042126  020040  044440
14151  070634  046114  043505  006514
14152  070642     012
14153  070643     040  020040  020040          .ASCIZ  /       NO      ADDRESS DATA    DATA    FUNCTN/
14154  070650  020040  047040  020117
14155  070656  020040  020040  040440
14156  070664  042104  042522  051523
14157  070672  042040  052101  020101
14158  070700  020040  042040  052101
14159  070706  020101  020040  043040
14160  070714  047125  052103  000116
14161
14162  070722  041520  020040  020040  DH60:   .ASCII  /PC      TEST    REG     GOOD    RECVD   MODFING/<15><12>
14163  070730  020040  042524  052123
14164  070736  020040  020040  042522
14165  070744  020107  020040  020040
14166  070752  047507  042117  020040
14167  070760  020040  042522  053103
14168  070766  020104  020040  047515
14169  070774  043104  047111  006507
14170  071002     012
14171  071003     040  020040  020040          .ASCIZ  /       NO      ADDRESS DATA    DATA    REG/
14172  071010  020040  047040  020117
14173  071016  020040  020040  040440
14174  071024  042104  042522  051523
14175  071032  042040  052101  020101
14176  071040  020040  042040  052101
14177  071046  020101  020040  051040
14178  071054  043505     000
14179  071057     120  020103  020040  DH61:   .ASCII  /PC      TEST    PC OF   RHCS1/<15><12>
14180  071064  020040  052040  051505
14181  071072  020124  020040  050040
14182  071100  020103  043117  020040
14183  071106  051040  041510  030523
14184  071114  005015
14185  071116  020040  020040  020040          .ASCIZ  /       NO      JSR     WAS/
14186  071124  020040  047516  020040
14187  071132  020040  020040  051512
14188  071140  020122  020040  020040
14189  071146  040527  000123
14190  071152  041520  020040  020040  DH62:   .ASCII  /PC      TEST    PC OF   RHDS1/<15><12>
14191  071160  020040  042524  052123
14192  071166  020040  020040  041520
14193  071174  047440  020106  020040
14194  071202  044122  051504  006461
14195  071210     012
14196  071211     040  020040  020040          .ASCIZ  /       NO      JSR     WAS/
14197  071216  020040  047040  020117
14198  071224  020040  020040  045040
14199  071232  051123  020040  020040
14200  071240  053440  051501     000
14201  071245     120  020103  020040  DH65:   .ASCII  /PC      TEST    CONT    CONT    CONT    /<15><12>
14202  071252  020040  052040  051505
14203  071260  020124  020040  041440
14204  071266  047117  020124  020040
```

```
14205  071274  041440  047117  020124
14206  071302  020040  041440  047117
14207  071310  020124  020040  006440
14208  071316     012
14209  071317     040  020040  020040          .ASCIZ  /       NO      RHCS1   RHAS    RHDS1   /
14210  071324  020040  047040  020117
14211  071332  020040  020040  051040
14212  071340  041510  030523  020040
14213  071346  051040  040510  020123
14214  071354  020040  051040  042110
14215  071362  030523  020040  000040
14216  071370  041520  020040  020040  DH66:   .ASCIZ  /PC      TEST    RHDST   RHER1   RHER2   RHER3   RHCS1   RHCS2/
14217  071376  020040  042524  052123
14218  071404  020040  020040  044122
14219  071412  051504  020124  020040
14220  071420  044122  051105  020061
14221  071426  020040  044122  051105
14222  071434  020062  020040  044122
14223  071442  051105  020063  020040
14224  071450  044122  051503  020061
14225  071456  020040  044122  051503
14226  071464  000062
14227
14228  071466  041520  020040  020040  DH77:   .ASCII  /PC      TEST    PC OF   REG.    GOOD    BAD/<15><12>
14229  071474  020040  042524  052123
14230  071502  020040  020040  041520
14231  071510  047440  020106  020040
14232  071516  042522  027107  020040
14233  071524  020040  047507  042117
14234  071532  020040  020040  040502
14235  071540  006504     012
14236  071543     040  020040  020040          .ASCIZ  /       NO      JSR     ADDR    DATA    DATA/
14237  071550  020040  047040  020117
14238  071556  020040  020040  045040
14239  071564  051123  020040  020040
14240  071572  040440  042104  020122
14241  071600  020040  042040  052101
14242  071606  020101  020040  042040
14243  071614  052101  000101
14244
14245                                  .EVEN
14246
14247  071620  001116  004604  041754  DT1:    .WORD   $ERRPC,TSTNM,WAITPC,WAITBT,WAITRE,$BDDAT,CS1,0
14248  071626  041760  041756  001126
14249  071634  002362  000000
14250  071640  001116  004604  041754  DT4:    .WORD   $ERRPC,TSTNM,WAITPC,WAITBT,WAITRE,$BDDAT,WAITTM,0
14251  071646  041760  041756  001126
14252  071654  041762  000000
14253  071660  001116  004604  004600  DT5:    .WORD   $ERRPC,TSTNM,REGADR,$GDDAT,$BDDAT,0
14254  071666  001124  001126  000000
14255  071674  001116  004604  004600  DT6:    .WORD   $ERRPC,TSTNM,REGADR,$BDDAT,0
14256  071702  001126  000000
14257  071706  001116  004604  001200  DT7:    .WORD   $ERRPC,TSTNM,$TMP1,0
14258  071714  000000
14259
14260  071716  001116  004604  001122  DT10:   .WORD   $ERRPC,TSTNM,$BDADR,CS1,CS2,DS1,ER1,0
```

```
14261  071724  002362  002360  002404
14262  071732  002364  000000
14263  071736  001116  004604  002362  DT26:   .WORD   $ERRPC,TSTNM,CS1,CS2,DS1,ER1,ER2,ER3,0
14264  071744  002360  002404  002364
14265  071752  002370  002376  000000
14266  071760  001116  004604  004602  DT30:   .WORD   $ERRPC,TSTNM,ERWORD,$GDDAT,$BDDAT,0
14267  071766  001124  001126  000000
14268
14269  071774  001116  004604  004600  DT51:   .WORD   $ERRPC,TSTNM,REGADR,$GDDAT,$BDDAT,ILLEGL,0
14270  072002  001124  001126  002464
14271  072010  000000
14272
14273  072012  001116  004604  004600  DT60:   .WORD   $ERRPC,TSTNM,REGADR,$GDDAT,$BDDAT,$BDADR,0
14274  072020  001124  001126  001122
14275  072026  000000
14276  072030  001116  004604  041560  DT61:   .WORD   $ERRPC,TSTNM,PCJSR,$BDADR,0
14277  072036  001122  000000
14278  072042  001116  004604  041560  DT62:   .WORD   $ERRPC,TSTNM,PCJSR,$BDADR,0
14279  072050  001122  000000
14280  072054  001116  004604  002362  DT65:   .WORD   $ERRPC,TSTNM,CS1,AS,DS1,0
14281  072062  002400  002404  000000
14282  072070  001116  004604  002364  DT66:   .WORD   $ERRPC,TSTNM,ER1,ER2,ER3,CS1,CS2,0
14283  072076  002370  002376  002362
14284  072104  002360  000000
14285
14286  072110  001116  004604  041560  DT77:   .WORD   $ERRPC,TSTNM,PCJSR,REGADR,$GDDAT,$BDDAT,0
14287  072116  004600  001124  001126
14288  072124  000000
14289
14290  072126     000     000     000  DF1:    .BYTE   0,0,0,0,0,0,0
14291  072131     000     000     000
14292  072134     000
14293  072135     000     000     000  DF4:    .BYTE   0,0,0,0,0,1,0
14294  072140     000     000     001
14295  072143     000
14296  072144     000     000     000  DF5:    .BYTE   0,0,0,0,0
14297  072147     000     000
14298  072151     000     000     000  DF6:    .BYTE   0,0,0,0
14299  072154     000
14300  072155     000     000     000  DF7:    .BYTE   0,0,0
14301
14302  072160     000     000     000  DF10:   .BYTE   0,0,0,0,0,0,0
14303  072163     000     000     000
14304  072166     000
14305
14306  072167     000     000     000  DF26:   .BYTE   0,0,0,0,0,0,0,0
14307  072172     000     000     000
14308  072175     000     000
14309
14310  072177     000     000     000  DF30:   .BYTE   0,0,0,0,0
14311  072202     000     000
14312
14313  072204     000     000     000  DF51:   .BYTE   0,0,0,0,0,0
14314  072207     000     000     000
14315
14316  072212     000     000     000  DF60:   .BYTE   0,0,0,0,0,0
```

```
 14317  072215      000      000      000
 14318  072220      000      000      000  DF61:   .BYTE   0,0,0,0
 14319  072223      000
 14320  072224      000      000      000  DF62:   .BYTE   0,0,0,0
 14321  072227      000
 14322  072230      000      000      000  DF65:   .BYTE   0,0,0,0,0
 14323  072233      000      000
 14324  072235      000      000      000  DF66:   .BYTE   0,0,0,0,0,0,0,0
 14325  072240      000      000      000
 14326  072243      000      000
 14327
 14328  072245      000      000      000  DF77:   .BYTE   0,0,0,0,0,0
 14329  072250      000      000      000
 14330
 14331             072254                    .EVEN
 14332
 14333
 14334             000001                    .END
```

```
ACE    = 020000            1909#   3366    3433
ACL    = 000040            1907#
ACU    = 100000            1867#
ADDMOD   000204             800#   2317
ADT1MO   044644           11694   11750#
AOE    = 001000            1789#
AS       002400            2005#  11469*  12414   14280
ATA    = 100000            1777#   3348    3417    3858    4898    5197    5467    5587    9023    9275   10678   11164
ATABLE   004666            2074#   2614
ATTENT   004740            2098#   2551*   2614*   3368    3442    3850    4044    4228    4400    4701    4889    5220    5469
                           5489    5589    9014    9266   10682
ATO    = 000001            1809#
AT1    = 000002            1810#
AT2    = 000004            1811#
AT3    = 000010            1812#
AT4    = 000020            1813#
AT5    = 000040            1814#
AT6    = 000100            1815#
AT7    = 000200            1816#
A16    = 000400            1752#
A17    = 001000            1753#
BA       002356            1994#  12342
BA1    = 000010            1723#   3998    4019
BASECH   043660             800   11679#  11756
BEGIN    005012             799    2129#  11749   12696
BEGIN1   004762             803    2123#
BEGIN2   004776             801    2126#
BIRD     020230            4756    4778#
BIT0   = 000001             759#
BIT00  = 000001             749#    759
BIT01  = 000002             748#    758
BIT02  = 000004             747#    757
BIT03  = 000010             746#    756
BIT04  = 000020             745#    755
BIT05  = 000040             744#    754
BIT06  = 000100             743#    753
BIT07  = 000200             742#    752
BIT08  = 000400             741#    751   11804
BIT09  = 001000             740#    750    2469   11812   12294
BIT1   = 000002             758#
BIT10  = 002000             739#  12278
BIT11  = 004000             738#  11819
BIT12  = 010000             737#
BIT13  = 020000             736#  12285
BIT14  = 040000             735#  11790
BIT15  = 100000             734#   5594
BIT2   = 000004             757#
BIT3   = 000010             756#
BIT4   = 000020             755#  11251
BIT5   = 000040             754#
BIT6   = 000100             753#
BIT7   = 000200             752#
BIT8   = 000400             751#
BIT9   = 001000             750#
BPTVEC= 000014              766#
CA       002374            2003#  12406
```

```
CAT     020174          4736    4759#
CC      002416          2012#   12446
CHECK   041562          2934    3039    4175    4368    4997    5256    5311    11144#
CHECKC  041620          11148   11154#
CHECKT  041604          3229    3543    3682    3798    3898    4575    4796    4969    5022    5325    5497    5644    5797
                        5946    6108    6302    6467    6654    6808    7003    7168    7381    7571    7813    7980    8227
                        8398    8645    8809    9102    9360    9443    9602    9697    9816    9934    10063   10190   10327
                        10498   11150#
CHREG   042404          2976    3106    3330    3343    3354    3361    3401    3412    3421    3428    3435    3853    3991
                        4002    4012    4067    4290    4301    4462    4473    4893    4900    5202    5212    5455    5462
                        5575    5582    9018    9025    9270    9277    10552   10654   10666   10673   10819   11389#
CKSWR = 104407          11789   12270   12293   12653#
CLAREA  041374          5743    5762    5911    6074    6248    6253    6263    6268    6415    6420    6428    6433    6610
                        6620    6766    6774    6949    6954    6964    6969    7116    7121    7129    7134    7325    7330
                        7340    7345    7517    7522    7530    7535    7730    7747    7766    7779    7927    7945    8143
                        8160    8179    8192    8347    8364    8591    8610    8750    8768    8773    9057    9072    9077
                        9314    9330    9335    9497    9502    9507    9512    9517    9547    9560    9568    9977    9982
                        9987    9992    9997    10026   10031   10238   10243   10248   10253   10258   10286   10291   10474
                        10590   10719   10726   11042#
CLDISK  041524          2547    2699    2734    2773    2833    2881    2930    3035    3161    3180    3225    3393    3539
                        3590    3678    3749    3894    4130    4171    4364    4527    4546    4571    4749    4768    4792
                        4965    4993    5307    5321    5614    5640    5672    5725    5907    6067    6241    6411    6603
                        6762    6942    7112    7309    7513    7714    7923    8127    8343    8526    8573    8746    8942
                        9166    9310    9439    9479    9666    9784    9930    9970    10179   10231   10467   10586   10713
                        10989   11122#
CLR   = 000040          1725#   11127
COMPAR  043542          5891    6037    6196    6394    6556    6746    6897    7095    7257    7496    7660    7907    8074
                        8327    8488    8898    9133    9391    9764    10775   11639#
COMREG  042512          2987    3118    3375    3449    3728    3865    4079    4313    4485    4715    4910    5227    5476
                        5598    5870    6017    6179    6375    6538    6727    6879    7076    7239    7476    7641    7886
                        8053    8306    8467    8727    8880    9034    9286    9646    9743    9887    10134   10398   10562
                        10690   10830   11456#
CPHALT  067033          2936    3041    3231    3545    3684    3800    3900    4177    4370    4577    4798    4971    4999
                        5313    5646    5799    5948    6110    6304    6469    6656    6810    7005    7170    7383    7573
                        7815    7982    8229    8400    8647    8811    9104    9362    9445    9604    9699    9818    9936
                        10065   10192   10329   10500   13983#
CR    = 000015          674#    11964   11974
CRLF  = 000200          675#    11935   11974
CSF   = 000002          1854#
CSU   = 000010          1856#
CS1     002362          1998#   3317    12358   14247   14260   14263   14280   14282
CS2     002360          1995#   3290    3600    12350   14260   14263   14282
DB      002352          1992#
DCK   = 100000          1795#
DCL   = 000100          1908#
DCLEAR  002430          2024#   3960
DDISP = 177570          681#    883     2161
DE1   = 000040          1767#
DFF20 = 000002          1763#
DFF5  = 000001          1762#
DF1     072126          943     955     966     14290#
DF10    072160          1020    1034    1048    1064    14302#
DF26    072167          1183    1660    1672    1684    1696    14306#
DF30    072177          1204    1223    1239    1256    1317    1351    1375    1411    1447    1556    1567    1598    1617
                        14310#
DF4     072135          979     14293#
```

```
DF5     072144          989    1074    1086    1100    1114    1126    1134    1141    1151    1160    1171    1194    1215
                       1233    1249    1267    1278    1292    1304    1329    1342    1364    1404    1423    1435    1460
                       1503    1515    1581    1591    1609    1629    1648   14296#
DF51    072204         1390   14313#
DF6     072151          999   14298#
DF60    072212         1477   14316#
DF61    072220         1486   14318#
DF62    072224         1495   14320#
DF65    072230         1527   14322#
DF66    072235         1545   14324#
DF7     072155         1007   14300#
DF77    072245         1639   14328#
DH1     067273          936     948     960   14017#
DH10    070071         1012    1026    1040    1056   14088#
DH26    070250         1175    1653    1665    1677    1689   14107#
DH30    070447         1199    1221    1237    1254    1315    1349    1373    1409    1445    1554    1565    1596    1615
                      14129#
DH4     067452          973   14037#
DH5     067613          984    1069    1081    1095    1109    1121    1132    1139    1149    1158    1169    1192    1213
                       1231    1247    1265    1276    1290    1302    1327    1340    1362    1402    1421    1433    1458
                       1501    1513    1576    1589    1607    1627    1643   14055#
DH51    070563         1384   14144#
DH6     067732          995   14070#
DH60    070722         1471   14162#
DH61    071057         1481   14179#
DH62    071152         1490   14190#
DH65    071245         1521   14201#
DH66    071370         1537   14216#
DH7     070031         1004   14081#
DH77    071466         1633   14228#
DIGB  = 000004         1764#
DISPLA  001142          883#    2161*   2169*  11833*  12277*
DISPRE  000174          782#    2169
DLT   = 100000         1735#
DL64  = 000020         1766#
DMD   = 000001         1799#    3162    3181    3252    3440    4528    4547    4750    4769   10969
DOG     017420         4513    4537#
DPR   = 000400         1770#    2898    3095    3315    4681    4873    5160    5191    5441    5561    5843    5990    6152
                       6348    6511    6700    6852    7049    7212    7426    7614    7859    8026    8273    8442    8693
                       8853    8997    9249    9860   10107   10372   11165
DRY   = 000200         1769#    2900    3095    3319    4655    4855    5197    5415    5535    5663    5686    8567    8979
                       9192    9231    9474    9964   10225   11165
DST     002366         2000#   12398
DSWR  = 177570          680#     882    2160
DS1     002404         2007#    3315    3319   10972   10984   12366   14260   14263   14280
DT      002406         2008#    2684*   4055   12470
DTE   = 010000         1792#
DTSY  = 001000         1805#
DT1     071620          942     954     965   14247#
DT10    071716         1019    1033    1047    1063   14260#
DT26    071736         1182    1659    1671    1683    1695   14263#
DT30    071760         1203    1222    1238    1255    1316    1350    1374    1410    1446    1555    1566    1597    1616
                      14266#
DT4     071640          978   14250#
DT5     071660          988    1073    1085    1099    1113    1125    1133    1140    1150    1159    1170    1193    1214
                       1232    1248    1266    1277    1291    1303    1328    1341    1363    1403    1422    1434    1459
```

```
                        1502    1514    1580    1590    1608    1628    1647   14253#
DT51    071774          1389   14269#
DT6     071674           998   14255#
DT60    072012          1476   14273#
DT61    072030          1485   14276#
DT62    072042          1494   14278#
DT65    072054          1526   14280#
DT66    072070          1544   14282#
DT7     071706          1006   14257#
DT77    072110          1638   14286#
DVA   = 004000          1755#   2755    2911    3086    3317    4664    4864    5141    5424    5544    5834    5981    6143
                        6339    6502    6691    6843    7040    7203    7417    7605    7850    8017    8264    8433    8684
                        8844    8988    9240    9851   10098   10363   11156
ECM   = 000100          1786#
EC1   = 004000          1880#   4297    4469    5594    5782    5930    5931    5932    6092    6093    6094    6287    6451
                        6452    6453    6639    6792    6793    6794    6988    7152    7153    7154    7364    7553    7554
                        7555    7799    7964    7965    7966    8212    8382    8383    8384    8630    8792    8793    8794
                        9095    9096    9097    9353    9354    9355    9587    9681    9682    9683    9799    9800    9801
                       10048   10049   10050   10308   10309   10310   10492   10608   10743   10744   10745
EC1     002412          2010#  12454
EC2     002414          2011#  12462
EMTVEC= 000030           769#   2145*   2146*   2295*   2338*
EM1     051166           934   12713#
EM10    051706          1010   12773#
EM100   066374          1642   13930#
EM101   066456          1651   13939#
EM102   066540          1663   13948#
EM103   066617          1675   13956#
EM104   066703          1687   13965#
EM11    051755          1023   12780#
EM12    052062          1037   12792#
EM13    052170          1051   12804#
EM14    052312          1067   12818#
EM15    052365          1077   12826#
EM16    052526          1089   12844#
EM17    052654          1103   12860#
EM2     051215           946   12717#
EM20    053011          1117   12878#
EM21    053155          1130   12896#
EM22    053230          1137   12904#
EM23    053303          1144   12912#
EM24    053446          1154   12930#
EM25    053611          1163   12948#
EM26    053773          1174   12968#
EM27    054063          1186   12978#
EM3     051304           958   12727#
EM30    054301          1197   13005#
EM31    054361          1207   13014#
EM32    054576          1218   13040#
EM33    054705          1226   13052#
EM34    055107          1236   13076#
EM35    055133          1242   13080#
EM36    055351          1252   13105#
EM37    055426          1259   13113#
EM4     051365           969   12736#
EM40    055643          1270   13139#
```

```
EM41    056052            1281   13163#
EM42    056333            1295   13194#
EM43    056611            1307   13225#
EM44    057153            1320   13265#
EM45    057445            1332   13298#
EM46    057741            1345   13331#
EM47    060054            1354   13344#
EM5     051476             982   12749#
EM50    060336            1367   13376#
EM51    060555            1378   13402#
EM52    061022            1394   13431#
EM53    061277            1407   13461#
EM54    061365            1414   13471#
EM55    061632            1426   13500#
EM56    062120            1438   13532#
EM57    062330            1450   13556#
EM6     051550             992   12756#
EM60    062622            1463   13590#
EM61    063233            1480   13637#
EM62    063314            1489   13646#
EM63    063375            1499   13655#
EM64    063451            1507   13663#
EM65    063670            1520   13688#
EM66    063712            1531   13691#
EM67    064104            1547   13712#
EM7     051636            1002   12765#
EM70    064374            1558   13745#
EM71    064602            1570   13770#
EM72    065043           13799#
EM73    065273            1594   13827#
EM74    065345            1583    1601   13835#
EM75    065611            1612   13864#
EM76    065714            1620   13876#
EM77    066151            1632   13904#
ERFLG$  004734            2094#  12273*
ERR     = 040000          1776#   3350    3419    5667   10680   10972   10984
ERRVEC= 000004             762#   2158    2159*   2170*   2297*   2305*   2340*  11795   11796*  11798*  11801*
ERWORD  004602            2053#  11652*  11658*  14266
ER1     002364            1999#  12374   14260   14263   14282
ER2     002370            2001#  12382   14263   14282
ER3     002376            2004#  12390   14263   14282
EXT1    = 000001          1834#
EXT10   = 000010          1837#
EXT2    = 000002          1835#
EXT20   = 000020          1838#
EXT4    = 000004          1836#
EXT40   = 000040          1839#
FEN     = 000200          1860#
FER     = 000020          1784#
FILL    042450           11419#
FILLRE  041426            3978    3981    4024    4027    4030    4033    4036    4039    4046    4051    4060    4063    4283
                          4286    4455    4458    4694    4697    4704    4707    4885    5592    5856    5859    5862    6003
                          6006    6009    6165    6168    6171    6361    6364    6367    6524    6527    6530    6713    6716
                          6719    6865    6868    6871    7062    7065    7068    7225    7228    7231    7462    7465    7468
                          7627    7630    7633    7872    7875    7878    8039    8042    8045    8286    8289    8292    8295
                          8298    8454    8457    8460    8706    8709    8713    8716    8719    8866    8869    8872    9010
```

```
                    9262   9632   9635   9638   9728   9731   9734   9873   9876   9879  10120  10123  10126
                   10385  10388  10391  10662  11070#
FINACC  004664      2068#  11226*
FINALA  004662      2067#  11227*
FIRST   004736      2095#  2183   2220*
FISH    017454      4534   4557#
FLHEAD  041350      5733   5755   7721   7738   7759   7772   7936   8134   8151   8172   8185   8355   8581
                    8600   8760   9065   9323   9486   9553  11013#
FMT22 = 010000      1881#  4295   4467   5594   5781   5782   5783   5930   5931   5932   6092   6093   6094
                    6286   6287   6288   6451   6452   6453   6638   6639   6640   6792   6793   6794   6987
                    6988   6989   7152   7153   7154   7363   7364   7365   7553   7554   7555   7798   7799
                    7800   7964   7965   7966   8211   8212   8213   8382   8383   8384   8629   8630   8631
                    8792   8793   8794   9095   9096   9097   9353   9354   9355   9586   9587   9588   9681
                    9682   9683   9799   9800   9801  10048  10049  10050  10308  10309  10310  10491  10492
                   10493  10607  10608  10609  10743  10744  10745
FUTABL  002422      2020#
GNS   = ****** U     781   2191   2195   2200   2204   2208   2213   2217   2272   2314   2329   2336   2390
                    2417   2421   2425   2429   2486   2492   2498   2575   2579   2617   2623   2629   2662
                    2668   2674   2704   2708   2712   2718   2855   2861   2887   2893   3276   3282   3302
                    3308   3552   3556   3570   3576   3580   3607   3613   3649   3653   4150   4161   4984
                    4988   5179   5245   5251   7447   7455  10540  10546  10807  10813  10870  10876  11524
                   11530  11537  11541  11545  11552  11556  11682  11688  11705  11711  11717  11723  11729
                   11733  11737  11741  11747  11753  11764  12332  12340  12348  12356  12364  12372  12380
                   12388  12396  12404  12412  12420  12428  12436  12444  12452  12460  12468  12476  12645
                   12646  12647  12648  12649  12651  12653  12654  12655  12656  12657
GO    = 000001      1749#  2960   3065   3473   3631   3710   3825   3961   4253   4266   4425   4438   4645
                    4664   4845   4864   5006   5121   5141   5207   5267   5404   5424   5524   5544   5655
                    5678   5812   5834   5961   5981   6123   6143   6317   6339   6482   6502   6669   6691
                    6823   6843   7018   7040   7183   7203   7395   7417   7585   7605   7828   7850   7995
                    8017   8242   8264   8413   8433   8557   8660   8684   8824   8844   8967   8988   9117
                    9181   9219   9240   9375   9463   9617   9712   9831   9851   9955  10078  10098  10216
                   10343  10363  10513  10631  10759  10977  11251
GRV   = 000010      1765#
GTSWR = 104406      2227  12651#
HCE   = 000200      1787#
HCI   = 002000      1879#  4299   4471   5594   5783   5932   6094   6288   6453   6640   6794   6989   7154
                    7365   7555   7800   7966   8213   8384   8631   8794   9097   9355   9588   9683   9801
                   10050  10310  10493  10609  10745
HCRC  = 000400      1788#
HT    = 000011       672#  11933  11974
HT.   = 003323      2968#  2973   3075#  3080   3482#  3487   3640#  3645   3718#  3723   3833#  3838   3969#
                    3974   4274#  4279   4446#  4451   4653#  4658   4853#  4858   5014#  5019   5276#  5281
                    5413#  5418   5533#  5538   5661#  5666   5684#  5689   5823#  5828   5970#  5975   6132#
                    6137   6328#  6333   6491#  6496   6680#  6685   6832#  6837   7029#  7034   7192#  7197
                    7406#  7411   7594#  7599   7839#  7844   8006#  8011   8253#  8258   8422#  8427   8565#
                    8570   8673#  8678   8833#  8838   8977#  8982   9124#  9129   9190#  9195   9229#  9234
                    9382#  9387   9472#  9477   9624#  9629   9719#  9724   9840#  9845   9962#  9967  10087#
                   10092  10223#  10228  10352#  10357  10521#  10526  10638#  10643  10766#  10771
IAE   = 002000      1790#
IE    = 000100      1750#  2744   2783   4266   4438   4645   4664   4668   4669   4845   4864   5145   5146
                    5404   5424   5428   5429   5524   5544   5548   5549   5655   5678   5812   5834   5961
                    5981   6123   6143   6317   6339   6482   6502   6669   6691   6823   6843   7018   7040
                    7183   7203   7395   7417   7585   7605   7828   7850   7995   8017   8242   8264   8413
                    8433   8557   8660   8684   8824   8844   8967   8988   9117   9181   9219   9240   9375
                    9463   9617   9712   9831   9851   9955  10078  10098  10216  10343  10363  10513  10631
                   10759  11277  11289  11357  11370
```

```
ILF    = 000001          1780#
ILLEGL  002464           2038#  14269
ILR    = 000002          1781#
INUNIT  004752           2105#  3561*  3619
IOTVEC= 000020            767#  2143*  2144*
IR     = 000100          1726#  2870   3996   4017   10650  10651
IXE    = 004000          1864#
KIPAR0= 172340            844#
KIPAR1= 172342            845#
KIPAR2= 172344            846#
KIPAR3= 172346            847#
KIPAR4= 172350            848#
KIPAR5= 172352            849#
KIPAR6= 172354            850#
KIPAR7= 172356            851#
KIPDR0= 172300            833#
KIPDR1= 172302            834#
KIPDR2= 172304            835#
KIPDR3= 172306            836#
KIPDR4= 172310            837#
KIPDR5= 172312            838#
KIPDR6= 172314            839#
KIPDR7= 172316            840#
LA      002420           2013#  12438
LBT    = 002000          1772#  11164
LF     = 000012           673#  11968  11974
LT.    = 000105          2968#   2973   3075#  3080   3482#  3487   3640#  3645   3718#  3723   3833#  3838   3969#
                         3974    4274#  4279   4446#  4451   4653#  4658   4853#  4858   5014#  5019   5276#  5281
                         5413#   5418   5533#  5538   5661#  5666   5684#  5689   5823#  5828   5970#  5975   6132#
                         6137    6328#  6333   6491#  6496   6680#  6685   6832#  6837   7029#  7034   7192#  7197
                         7406#   7411   7594#  7599   7839#  7844   8006#  8011   8253#  8258   8422#  8427   8565#
                         8570    8673#  8678   8833#  8838   8977#  8982   9124#  9129   9190#  9195   9229#  9234
                         9382#   9387   9472#  9477   9624#  9629   9719#  9724   9840#  9845   9962#  9967   10087#
                        10092   10223#  10228  10352#  10357  10521#  10526  10638#  10643  10766#  10771
MAKECY  041174           3163   3182   4529   4548   4751   4770   10961#
MCLK   = 000002          1800#
MCPE   = 020000          1756#  2909   3341   3410
MHS    = 001000          1862#
MINX   = 000004          1801#
MMVEC  = 000250           822#
MOL    = 010000          1774#  2700   2714   2864   2896   3095   4156   4168   4681   4873   4980   5162   5163
                         5193   5194   5441   5561   5843   5990   6152   6348   6511   6700   6852   7049   7212
                         7426   7614   7859   8026   8273   8442   8693   8853   8997   9249   9860   10107  10372
                        11165
MPE    = 000400          1728#
MR      002402           2006#  12430
MRD    = 000020          1803#
MSE    = 000020          1857#
MSTCK  = 000010          1802#
MWR    = 000040          1804#
MXF    = 001000          1729#  4000
NED    = 010000          1732#  3290   3359   3426   3600
NEM    = 004000          1731#  7442
NHS    = 002000          1863#
NOPERA  002422           2021#  3688   3709   3803   3824
NOPUSH  004724           2088#  2123*  2127*  2130*  2842   3237   3525   4137   4951   10453
```

```
NOUNIT  004720          2084#  2446*  2506*  2512   2518   2585*  10894*
NOUSE   066766          2475  13974#
NUNIT   004722          2086#  2518*  2519*  3844   4020   4665   5142   5209   5425   5545
OCYL  = 100000          1911#
OF      002372          2002#  12422
OFREV = 000200          1878#
OFSET   041510          5649  11117#
OFSETC  002454          2034#  5383   5403   5423   5654  11118
OFSTVL  004610          2060#  5315*  5316*  5353   5618*  5619   5621*  5622
OF100 = 000004          1873#
OF200 = 000010          1874#
OF25  = 000001          1871#
OF400 = 000020          1875#
OF50  = 000002          1872#
OF800 = 000040          1876#
OPERSE  042650          11519#  12016  12099
OPI   = 020000          1793#
OR    = 000200          1727#  10650  10651
PAR   = 000010          1783#  4072
PCJSR   041560          10963*  10964*  11142#  11144*  11145*  11150*  11151*  14276  14278  14286
PCLBUF  002346          1972#  11250*
PCLCSR  002344          1971#  11224*  11251*
PCLCTR  002350          1973#  11225
PGE   = 002000          1730#
PIP   = 020000          1775#  4681   4873   5160   5191   5441   5561   8997   9249
PIRQ  = 177772           679#
PIRQVE= 000240           773#
PKACK   002460          2036#  2938   2959   3043   3064   3085   3472   3630   5005   5266
PLU   = 020000          1866#
PRE   = 000020          1906#
PRITEM  047342          2263*  3010*  3586*  4336*  5254*  10848*  10893*  11521*  12319#  12481*  12498  12502*
PROG  = 001000          1771#
PR0   = 000000           696#
PR1   = 000040           697#
PR2   = 000100           698#
PR3   = 000140           699#
PR4   = 000200           700#
PR5   = 000240           701#
PR6   = 000300           702#
PR7   = 000340           703#
PS    = 177776           676#   677   2175*  2743*  2782*  2837*  10866*  11520*
PSEL  = 002000          1754#  7369   7395   7417   7559   7585   7605   10316  10343  10363
PSU   = 000001          1903#
PSW   = 177776           677#
PUTREG  043436          3289   3314   3599  10971  10983  11468  11576#
PWRVEC= 000024           768#  2149*  2150*  12663*  12664*  12673*  12679*  12691*  12692*
RA      000200           799#  11743
RDCHR = 104410         12174  12654#
RDLIN = 104411         12221  12655#
RDOCT = 104412          2274  11547  11558  11691  11713  12656#
RDY   = 000200          1751#  2744   2755   2783   2911   3086   3335   3720   3835   3971   4448   4664   4864
                        5132   5141   5424   5544   5825   5972   6134   6330   6493   6682   6834   7031   7194
                        7408   7596   7841   8008   8255   8424   8675   8835   8988   9126   9240   9384   9626
                        9721   9842  10089  10354  10523  10640  10768  11156
READAT  002446          2031#  6095   6122   6142   6454   6481   6501   6795   6822   6842   7155   7182   7202
                        7556   7584   7604  10746  10758
```

```
READIN  002462  2037#  4229  4252  4265  4401  4424  4437
RECALI  002426  2023#  4625  4644  4663  4825  4844  4863  5677  8556  9180
REFOR   002450  2032#  5933  5960  5980  7967  7994  8016  8385  8412  8432  8795  8823  8843
                9098   9116  9356  9374  9684  9711
REGADR  004600  2052#  2361* 2922* 3092* 3101* 4678* 4687* 4870* 4879* 5155* 5171* 5438* 5447*
                5558*  5567* 5840* 5849* 5987* 5996* 6149* 6158* 6345* 6354* 6508* 6517* 6697*
                6706*  6849* 6858* 7046* 7055* 7209* 7218* 7423* 7432* 7611* 7620* 7856* 7865*
                8023*  8032* 8270* 8279* 8439* 8448* 8690* 8699* 8850* 8859* 8994* 9003* 9246*
                9255*  9857* 9866* 10104* 10113* 10369* 10378* 10994* 11477* 14253 14255 14269 14273
                14286
REGSAV  047176  12272#
REGSA1  047204  2295   12274#
REINTO  003534  2046#  5756  5763  5892  5912  5926  6008  6039  6088  6170  6198  6264  6269
                6395   6416  6421  6447  6529  6558  6621  6747  6767  6788  6870  6899  6965
                6970   7096  7117  7122  7148  7230  7259  7341  7346  7497  7518  7523  7549
                7632   7662  7760  7767  7773  7780  7908  7928  7960  8044  8076  8173  8180
                8186   8193  8328  8348  8378  8459  8490  8751  8788  8871  8900  9058  9091
                9135   9315  9349  9393  9569  9677  9733  9766  10720 10739 10777
RELEAS  002432  2025#
RESVEC= 000010   763#
RETCL   002456  2035#  5502  5523  5543
RHAS    002316  1953#  2355  2392  4228* 4400* 5489*
RHBA    002274  1941#  3982  5860  6007  6169  6365  6528  6717  6869  7066  7229  7466  7631
                7876   8043  8290  8458  8710  8870  9636  9732  9877  10124 10389 10648 11609*
RHBAE   002340  1965#  2325
RHCA    002312  1951#  4037  4284  4456  8293  8717  10966* 11090* 11108* 11606*
RHCC    002334  1960#  2920  2922  2924  3951  4695  4886  8296  8714  9011  9263  10990 10994
                11226  11582
RHCS1   002300  1946#  2938* 2962* 3043* 3067* 3331  3402  3475* 3633* 3688* 3712* 3719  3803*
                3827*  3834  3963* 3970  4003  4068  4229* 4255* 4268* 4401* 4427* 4440* 4447
                4625*  4647* 4705  4825* 4847* 4901  5008* 5099* 5123* 5131  5203  5213  5269*
                5383*  5406* 5456  5502* 5526* 5576  5657* 5680* 5814* 5824  5963* 5971  6125*
                6133   6319* 6329  6484* 6492  6671* 6681  6825* 6833  7020* 7030  7185* 7193
                7369*  7397* 7407  7559* 7587* 7595  7830* 7840  7997* 8007  8244* 8254  8415*
                8423   8559* 8662* 8674  8826* 8834  8969* 9026  9119* 9125  9183* 9221* 9278
                9377*  9383  9465* 9619* 9625  9714* 9720  9833* 9841  9957* 10080* 10088 10218*
                10316* 10345* 10353 10515* 10522 10633* 10639 10655 10761* 10767 10968* 10977* 11092*
                11109* 11118* 11122 11277 11289 11357 11370 11615* 11684 11695 11698 11719
RHCS2   002276  1942#  2393  2452* 2593* 3355  3422  3992  4013  7442  10649 11123 11612*
RHCS3   002342  1966#
RHDB    002270  1939#  2300  11692
RHDST   002304  1948#  3288  3598  4028  4287  4459  5863  6010  6172  6368  6531  6720  6872
                7069   7232  7469  7634  7879  8046  8299  8461  8720  8873  9639  9735  9880
                10127  10392 10663 10967* 11091* 11607*
RHDS1   002322  1955#  2469  2969  2977  3076  3107  3344  3413  3483  3641  3854  4052  4275
                4302   4474  4654  4708  4854  4894  4978  5015  5192  5277  5414  5463  5534
                5583   5662  5667  5685  8566  8978  9019  9191  9230  9271  9473  9963  10224
                10553  10674 10820 11124
RHDT    002324  1956#  2453  2455  2459  2461  2464  2466  2494  2594  2596  2603  2605  2631
                2638   2640  2643  2645  2648  2650  2682  2684
RHEC1   002330  1958#  4061
RHEC2   002332  1959#  4064
RHER1   002302  1947#  2396  4025  10667 11125
RHER2   002306  1949#  4031
RHER3   002314  1952#  3362  3429  4040
RHLA    002336  1961#  11227
```

RHMR	002320	1954#	3162*	3181*	3252*	3436	4047	4528*	4547*	4750*	4769*	10969*		
RHOF	002310	1950#	4034	4291	4463	4698	5593	11117*	11613*					
RHSN	002326	1957#	2625	2681	2683									
RHWC	002272	1940#	2945	3050	3257	3695	3753	3810	3906	3979	4179	4236	4372	4408
		4580	4632	4801	4832	5028	5106	5330	5389	5509	5791	5857	5940	6004
		6102	6166	6296	6362	6461	6525	6648	6714	6802	6866	6997	7063	7162
		7226	7375	7463	7565	7628	7807	7873	7974	8040	8221	8287	8392	8455
		8639	8707	8803	8867	8952	9204	9596	9633	9691	9729	9810	9874	10057
		10121	10321	10386	10532	10616	10647	10798	11075	11394	11477	11580	11582	11608*
RH70	004750	2103#	2234*	2324*	2326*	3842	3987	4249	4421	7316	10186			
RH70CK	006150	2232#												
RMR =	000004	1782#												
RPTRP1	011526	2740	2754#											
RPTRP2	011634	2779	2793#											
RPVEC	002266	1704#	2177	2739	2778	2951*	3056*	3701*	3816*	3952*	4242*	4414*	4637*	4837*
		5112*	5395*	5515*	5803*	5952*	6114*	6308*	6473*	6660*	6814*	7009*	7174*	7387*
		7577*	7819*	7986*	8233*	8404*	8549*	8651*	8815*	8958*	9108*	9172*	9210*	9366*
		9454*	9608*	9703*	9822*	9946*	10069*	10207*	10333*	10504*	10622*	10750*	11707	11714*
		11725	11767*											
RPVECT	044734	2178	11761#	11767										
RP05	004746	2102#	2602*	2608*	3143	3214	3507							
RP06	004744	2101#	2592*	2599*	3141	3212	3505	4510	4733					
RP4VEC	004606	2055#	2262*	2951	3056	3701	3816	3952	4242	4414	4637	4837	5112	5395
		5515	5803	5952	6114	6308	6473	6660	6814	7009	7174	7387	7577	7819
		7986	8233	8404	8549	8651	8815	8958	9108	9172	9210	9366	9454	9608
		9703	9822	9946	10069	10207	10333	10504	10622	10750				
RTN	006642	2331	2338#											
RUN	043476	5771	5920	6083	6277	6442	6629	6783	6978	7143	7354	7544	7788	7954
		8201	8372	8619	8782	9085	9343	9576	9671	9789	10039	10299	10482	10598
		10734	11606#											
SAVER	041672	2944	3049	3256	3694	3809	4235	4407	4631	4831	5105	5388	5508	5790
		5939	6101	6295	6460	6647	6801	6996	7161	7374	7564	7806	7973	8220
		8391	8638	8802	8951	9203	9595	9690	9809	10056	10320	10531	10615	10797
		11189#												
SAVERE	004612	2063#	2946	2989	3051	3120	3258	3368*	3369*	3377	3442*	3451	3696	3730
		3811	3846*	3850*	3867	3951*	3984*	3985*	4022*	4045*	4055*	4057*	4081	4237
		4315	4409	4487	4633	4701*	4717	4833	4889*	4912	5107	5198*	5220*	5229
		5390	5469*	5478	5510	5589*	5600	5792	5872	5941	6019	6103	6181	6297
		6377	6462	6540	6649	6729	6803	6881	6998	7078	7163	7241	7376	7478
		7566	7643	7808	7888	7975	8055	8222	8308	8393	8469	8640	8729	8804
		8882	8953	9014*	9036	9205	9266*	9288	9597	9648	9692	9745	9811	9889
		10058	10136	10322	10400	10533	10564	10617	10647*	10648*	10651*	10652*	10682*	10692
		10799	10832	11076*	11397*	11399*	11469							
SC =	100000	1758#	2909	3337	3406	3846	4007	4022	4671	4905	5148	5217	5431	5460
		5551	5580	9030	9282	10659								
SC1 =	000100	1840#												
SC10 =	001000	1843#												
SC2 =	000200	1841#												
SC20 =	002000	1844#												
SC4 =	000400	1842#												
SEECOM	002452	2033#	8966	8987	9218	9239	9462	9954	10215	10968	11109			
SEEKCY	041474	8947	9199	9451	9943	10204	11108#							
SELECT	004726	2090#	2124*	2126*	2129*	2268	2522	2559	10882					
SELTST	010066	2411	2522#											
SERCH	002434	2026#	11092											
SKI =	040000	1910#												

```
SN      002410          2009#   2683*   4057    12478
SND1    006112          2186    2220#
SRCH    041454         11090#
SRO   = 177572           826#
SR1   = 177574           827#
SR2   = 177576           828#
SR3   = 172516           829#
STACK = 001000           667#   2141    2294    2352    2544    2732    2771    2830    3032    3158    3177    3222    3536
                        3675    3891    4127    4361    4524    4543    4568    4746    4765    4789    4962    5304    5637
                        5722    6064    6238    6600    6939    7306    7711    8124    8523    9163    9436    9927   10176
                       10464
START   005022          2125    2128    2133#
STKLMT= 177774           678#
ST1     032606          8536*   8579#   8920*   8933
ST10    033210          8544*   8767#   8928*
ST11    033234          8545*   8780#   8929*
ST12    033470          8885    8888#
ST13    033474          8886    8896#
ST14    033512          8902    8906#
ST15    033516          8903    8920#
ST16    033604          8931    8933#
ST17    033610          8932    8940#
ST18    034040          9039    9042#
ST19    034044          9040    9056#
ST2     032626          8537*   8590#   8921*
ST20    040414         10804#
ST25    035676          9808#
ST26    034230          9137    9141#
ST27    034234          9138    9146#
ST3     032640          8538*   8598#   8922*
ST4     032660          8539*   8609#   8923*
ST5     032672          8540*   8617#   8924*
ST6     033100          8541*   8542*   8712#   8925*   8926*
ST7     033146          8732    8734#
ST8     033152          8733    8744#
ST9     033170          8543*   8758#   8927*
SWR     001140           882#   2139    2160*   2162    2168*   2225    2232    2309    2386    2409    3517   11481   11661
                       11790   11804   11806   11812   11819   12019   12058   12113*  12278   12285   12290   12294   12321
                       12671   12684*
SWREG   000176           783#   2168    2225    12019   12058   12081
SW0   = 000001           731#
SW00  = 000001           721#    731
SW01  = 000002           720#    730
SW02  = 000004           719#    729
SW03  = 000010           718#    728
SW04  = 000020           717#    727
SW05  = 000040           716#    726
SW06  = 000100           715#    725    12323
SW07  = 000200           714#    724    11483   11663
SW08  = 000400           713#    723
SW09  = 001000           712#    722
SW1   = 000002           730#
SW10  = 002000           711#
SW11  = 004000           710#
SW12  = 010000           709#   2232
SW13  = 020000           708#   2309    2386    2409
```

```
SW14   = 040000            707#
SW15   = 100000            706#
SW2    = 000004            729#
SW3    = 000010            728#
SW4    = 000020            727#
SW5    = 000040            726#   3517
SW6    = 000100            725#
SW7    = 000200            724#
SW8    = 000400            723#
SW9    = 001000            722#
TBITVE= 000014             764#
TDF    = 000040           1858#
TESTAD  042646           11518#
TIMCNT  042222            2746   2785   11266  11271  11332#  11346  11351
TIME1   041724           11224#
TIME2   041736            2262  11226#
TKVEC  = 000060            771#  11996*  11997*
TMP0    004754            2112#
TMP1    004756            2113#  5635*  5690*
TMP4    004760            2114#  4667*  4668*  4670   5144*  5145*  5147   5161*  5162*  5164   5192*  5193*  5195
                          5427*  5428*  5430   5547*  5548*  5550
TN     = 000050            2544#  2545   2830#  2831   3032#  3033   3222#  3223   3536#  3537   3675#  3676   3891#
                           3892   4127#  4128   4361#  4362   4568#  4569   4789#  4790   4962#  4963   5304#  5305
                          5637#  5638   5722#  5723   6064#  6065   6238#  6239   6600#  6601   6939#  6940   7306#
                           7307   7711#  7712   8124#  8125   8523#  8524   9163#  9164   9436#  9437   9927#  9928
                          10176#  10177  10464#  10465
TOTALA  004742            2099#  2402*  2404*  3779   3932   4042   4209   4606   5078   5365
TPVEC  = 000064            772#
TRAPVE= 000034             770#  2147*  2148*  2176*
TRE    = 040000           1757#  2909   3339   3408   10661
TRK1   = 004000           1845#
TRK10  = 040000           1848#
TRK2   = 010000           1846#
TRK20  = 100000           1849#
TRK4   = 020000           1847#
TRP     006602            2323   2332#
TRTVEC= 000014              765#
TSTNM   004604            2054#  2293*  2353*  2382*  2545*  2698*  2731*  2770*  2831*  3033*  3159*  3178*  3223*
                          3537*  3676*  3892*  4128*  4362*  4525*  4544*  4569*  4747*  4766*  4790*  4963*  5305*
                          5638*  5723*  6065*  6239*  6601*  6940*  7307*  7712*  8125*  8524*  9164*  9437*  9928*
                         10177*  10465*  11526  14247  14250  14253  14255  14257  14260  14263  14266  14269  14273
                         14276  14278  14280  14282  14286
TST1    006314            2269   2281   2291#  10942
TST10   011644            2791   2829#
TST11   012424            2851   3024#
TST12   012706            3157#
TST13   012746            3148   3176#
TST14   013002            3169   3208#
TST15   014014            3218   3246   3324   3501#
TST16   015022            3145   3511   3519   3534   3674#
TST17   015460            3890#
TST2    006656            2350#
TST20   016246            4126#
TST21   017012            4146   4360#
TST22   017360            4523#
TST23   017420            4542#
```

```
TST24   017454          4567#
TST25   020134          4745#
TST26   020174          4764#
TST27   020230          4788#
TST3    006732          2358    2380#
TST30   020576          4946#
TST31   022112          4960    5302#
TST32   023154          5625    5634#
TST33   023326          5691    5721#
TST34   024214          6063#
TST35   024530          6237#
TST36   025440          6599#
TST37   026300          6938#
TST4    010102          2523    2541#   10891   10902
TST40   027210          7304#
TST41   030360          7318    7710#
TST42   031370          8123#
TST43   032420          8522#
TST44   034234          9162#
TST45   034740          9435#
TST46   036114          9926#
TST47   036630         10175#
TST5    011070          2697#
TST50   037376         10188   10448#
TST51   040622         10462   10864#
TST6    011434          2701    2729#
TST7    011544          2752    2756    2768#
TUF   = 000100          1859#
TYPDS = 104405          2474    2489    2505    2620    2858    2890    3279    3305    3573    3610    4154    4165    5248
                       10543   10810   10873   10879   10889   10931   12524   12649#
TYPE  = 104401          2189    2193    2198    2202    2206    2211    2215    2270    2312    2327    2334    2388    2415
                        2419    2423    2427    2472    2475    2484    2490    2496    2503    2573    2577    2615    2621
                        2627    2660    2666    2672    2702    2706    2710    2716    2853    2859    2885    2891    2936
                        3041    3231    3274    3280    3300    3306    3545    3550    3554    3568    3574    3578    3605
                        3611    3647    3651    3684    3800    3900    4148    4155    4159    4166    4177    4370    4577
                        4798    4971    4982    4986    4999    5177    5243    5249    5313    5646    5799    5948    6110
                        6304    6469    6656    6810    7005    7170    7383    7445    7449    7450    7453    7457    7458
                        7573    7815    7982    8229    8400    8647    8811    9104    9362    9445    9604    9699    9818
                        9936   10065   10192   10329   10500   10538   10544   10805   10811   10868   10874   10887   10890
                       10929   10932   11522   11528   11534   11535   11539   11543   11550   11554   11680   11686   11703
                       11709   11715   11721   11727   11731   11735   11739   11745   11751   11762   11895   11938   12013
                       12025   12079   12080   12083   12094   12104   12115   12134   12178   12181   12185   12250   12252
                       12280   12288   12330   12338   12346   12354   12362   12370   12378   12386   12394   12402   12410
                       12418   12426   12434   12442   12450   12458   12466   12474   12483   12505   12507   12510   12512
                       12527   12597   12645#  12693
TYPERR  050304         12326   12482#
TYPOC = 104402          2319    2495    2626    2632   11527   11533   11685   11708   11720   11726   11744   11766   12082
                       12335   12343   12351   12359   12367   12375   12383   12391   12399   12407   12415   12423   12431
                       12439   12447   12455   12463   12471   12479   12491   12520   12646#
TYPON = 104404         12648#
TYPOS = 104403         12647#
UBUSB   004732          2092#   7321*   7444*  10196
UNIB  = 000020          1724#
UNIT    004716          2083#   2276*   2282*   2517*   2524*   2553   2584*   2587   2619   2857   2869   2889   3278
                        3304    3561    3572    3587*   3588*   3609   3619   3985   4153   4164   5247  10542  10809
                       10872   10896   10901*  11128   11610
```

```
UNITS   004676          2082#   2440    2444    2517    2566    10897
UNITSL  004730          2091#   2277*   2282    2524
UNLOAD  002424          2022#   5099    5120    5140
UNS    = 040000         1794#
UPE    = 020000         1733#
US1    = 000001         1720#
US2    = 000002         1721#
US4    = 000004         1722#
UWR    = 000010         1905#
VUF    = 000002         1904#
VU30   = 010000         1865#
VV     = 000100         1768#   2902    2970    2981    3077    3095    3111    3352    3484    3642    4276    4306    4478
                        4681    4873    4980    5016    5162    5163    5193    5194    5278    5441    5561    5843    5990
                        6152    6348    6511    6700    6852    7049    7212    7426    7614    7859    8026    8273    8442
                        8693    8853    8997    9249    9860    10107   10372   11147   11165
WAITBT  041760          11247#  11260*  11267   11272   11341*  11347   11352   14247   14250
WAITPC  041754          11245#  11257*  11258*  11338*  11339*  14247   14250
WAITRE  041756          11246#  11259*  11267   11272   11276   11288   11340*  11347   11352   11356   11372   14247   14250
WAITTM  041762          11225*  11248#  11249*  11296   11306   14250
WAIT.P  041764          11249#
WAIT.T  042224          11334#  12657
WAT    = 104413          2968    3075    3482    3640    3718    3833    3969    4274    4446    4653    4853    5014    5130
                         5276    5413    5533    5661    5684    5823    5970    6132    6328    6491    6680    6832    7029
                         7192    7406    7594    7839    8006    8253    8422    8565    8673    8833    8977    9124    9190
                         9229    9382    9472    9624    9719    9840    9962    10087   10223   10352   10521   10638   10766
                        12657#
WC      002354          1993#   2990    3121    3378    3452    3731    3868    4082    4316    4488    4718    4913    5230
                         5479    5601    5873    6020    6182    6378    6541    6730    6882    7079    7242    7479    7644
                         7889    8056    8309    8470    8730    8883    9037    9289    9649    9746    9890    10137   10401
                        10565   10693   10833   11581   12334
WCE    = 040000         1734#
WCF    = 000040         1785#
WCU    = 000001         1853#
WLE    = 004000         1791#   10671
WRCHDT  002440          2028#   9802    9830    9850
WRCHEK  002436          2027#   10051   10077   10097   10311   10342   10362
WRFROM  002470          2045#   5734    5744    5777    5861    5893    6038    6075    6197    6249    6254    6282    6366
                         6396    6429    6434    6557    6611    6634    6718    6748    6775    6898    6950    6955    6983
                         7067    7097    7130    7135    7258    7326    7331    7359    7467    7498    7531    7536    7661
                         7722    7731    7739    7748    7794    7877    7909    7937    7946    8075    8135    8144    8152
                         8161    8207    8291    8329    8356    8365    8489    8582    8592    8601    8611    8625    8711
                         8761    8769    8774    8899    9066    9073    9078    9134    9324    9331    9336    9392    9487
                         9498    9503    9508    9513    9518    9528    9548    9554    9561    9582    9637    9765    9795
                         9878    9978    9983    9988    9993    9998    10008   10027   10032   10044   10125   10239   10244
                        10249   10254   10259   10269   10287   10292   10304   10390   10475   10487   10591   10603   10727
                        10776
WRIDAT  002442          2029#   6289    6316    6338    6641    6668    6690    6990    7017    7039    7366    7394    7416
                        10494   10512   10610   10630
WRIFOR  002444          2030#   5784    5811    5833    7801    7827    7849    8214    8241    8263    8632    8659    8683
                         9589    9616
WRL    = 004000         1773#   10557   10824
WRU    = 000400         1861#
WSU    = 000004         1855#
XE2     007464          2407    2438#
$AUTOB  001134           879#   2229*   12072   12201
$BDADR  001122           874#   2866*   2874*   2904*   2914*   11159*  11167*  14260   14273   14276   14278
```

```
$BDDAT  001126   876# 2359* 2924* 3091* 3100* 4677* 4686* 4869* 4878* 5154* 5170* 5437* 5446*
                5557* 5566* 5839* 5848* 5986* 5995* 6148* 6157* 6344* 6353* 6507* 6516* 6696*
                6705* 6848* 6857* 7045* 7054* 7208* 7217* 7422* 7431* 7610* 7619* 7855* 7864*
                8022* 8031* 8269* 8278* 8438* 8447* 8689* 8698* 8849* 8858* 8993* 9002* 9245*
                9254* 9856* 9865* 10103* 10112* 10368* 10377* 10990* 10991 11276* 11288* 11356* 11372*
                11476* 11657* 14247 14250 14253 14255 14266 14269 14273 14286
$BELL    001216   908# 12025 12194 12280 12302
$CHARC   045710  11940* 11950* 11957 11966* 11971#
$CKSWR   046222  12058# 12653
$CMTAG   001100   862# 2136 2137 2145 2151 2152
$CM1   = 000006   894#  895#  896#  897#  898#  899#  900#
$CM2   = 000014   894#  895#  896#  897#  898#  899#  900#
$CM3   = 000006   892#  894
$CM4   = 000006   900#  901#  902#  903#  904#  905#  906#
$CNTLC   046773  12013 12094 12194#
$CNTLG   047005  12079 12196#
$CNTLU   047000  12104 12195#
$CRLF    001223   910# 2472 2503 4155 4166 7449 7450 7457 7458 11534 11939 11974 12115
                12194 12255 12288 12302 12483 12507 12512
$DBLK    045464  11861 11895 11903#
$DOAGN   041150  10925 10934 10940#
$DTBL    045454  11864 11899#
$ENDAD   041140    791 10936#
$ENDCT   041106  10927#
$ENDMG   041157  10887 10929 10944#
$ENULL   041154  10890 10932 10943#
$EOP     041052   2321  2432  2514 10895 10917#
$EOPCT   041100  10924# 10928
$ERFLG   001103   865# 11779 11808 11810 11816* 11837 12275* 12302
$ERMAX   001115   871# 2153* 11810 11832* 11837
$ERROR   047174  2145  2338 12269#
$ERRPC   001116   872# 12282* 12283* 12284 12302 12489 14247 14250 14253 14255 14257 14260 14263
                14266 14269 14273 14276 14278 14280 14282 14286
$ERRTB   001226   927# 12497
$ERRTY   047344  12287 12321#
$ERTTL   001112   869# 10878 10880* 12281* 12302
$ESCAP   001214   907# 2152* 11831* 12297 12299 12302
$FILLC   001156   890# 11943 11974
$FILLS   001155   889# 11974
$GDADR   001120   873#
$GDDAT   001124   875# 2360* 2923* 3087* 3096* 4673* 4682* 4865* 4874* 5150* 5166* 5433* 5442*
                5553* 5562* 5835* 5844* 5982* 5991* 6144* 6153* 6340* 6349* 6503* 6512* 6692*
                6701* 6844* 6853* 7041* 7050* 7204* 7213* 7418* 7427* 7606* 7615* 7851* 7860*
                8018* 8027* 8265* 8274* 8434* 8443* 8685* 8694* 8845* 8854* 9989* 8998* 9241*
                9250* 9852* 9861* 10099* 10108* 10364* 10373* 10993* 11475* 11656* 14253 14266 14269
                14273 14286
$GET42   041130  10933#
$GTSWR   046312  12080# 12651
$HD    = 000000    619
$HIOCT   047172  12243* 12254#
$ICNT    001104   866# 11823* 11824 11826* 11836
$ILLUP   051150  12663 12679 12698#
$INTAG   001135   880# 12077* 12096 12116 12201
$ITEMB   001114   870# 12284* 12302 12486
$LF      001224   911# 11974 12185 12194 12255 12302
$LPADR   001106   867# 2154* 2543* 5303* 11549* 11560 11814* 11829* 11834 11836
```

```
$LPERR  001110          868#  2155*  10583*  11532  11559*  11814  11830*  11836  12296
$MAIL = ****** U       2172   2225   11829   11927   12290
$MREW   047023        12083  12199#
$MSWR   047012        12080  12197#
$MXCNT  045246        11827  11836#
$NULL   001154          888#  11945  11974
$NWTST= 000001         2285#  2287   2344#   2346   2367#   2369   2528#   2530   2689#   2691   2721#   2723   2762#
                       2764   2802#  2804    3015#  3017    3154#  3173#   3190#  3192    3492#  3494    3666#  3668
                       3883#  3885   4098#   4100   4339#   4341   4520#   4539#  4558#   4560   4742#   4761#  4779#
                       4781   4925#  4927    5283#  5285    5628#  5630    5695#  5697    6052#  6054    6209#  6211
                       6571#  6573   6910#   6912   7273#   7275   7680#   7682   8089#   8091   8504#   8506   9150#
                       9152   9410#  9412    9906#  9908    10153# 10155   10422# 10424   10853# 10855
$OCNT   050714        12569* 12598* 12611#
$OMODE  050716        12564* 12568* 12573   12576* 12587*  12613#
$OVER   045232        11791  11807  11815   11825  11833#
$PASS   001100          863#  2846   3241    3529   4141    4955   10457  10886*  10888  10921*  10922*  10930  10943
                      11821  11837
$POWER  051156        12694  12701#
$PWRAD  051144        12696#
$PWRDN  051004         2149  12663#  12691
$PWRMG  051140        12694#
$PWRUP  051056        12673  12679#
$QUES   001222          909#  11974  12134   12178   12194   12252  12255   12302
$RDCHR  046564        12147# 12654
$RDDEC= ****** U      12657
$RDLIN  046654        12170# 12655
$RDOCT  047034        12216# 12656
$RDSZ = 000011        12163#
$REGAD  001160          892#
$REG0   001162          894#
$REG1   001164          895#
$REG2   001166          896#
$REG3   001170          897#
$REG4   001172          898#
$REG5   001174          899#
$RTNAD  041152        10942#
$R2A  = ****** U      12657
$SAVRE= ****** U      12657
$SAVR6  051154        12672* 12680  12681*  12682*  12700#
$SCOPE  044776         2143  11788#
$SETUP= 000117         2134#  2142   2143    2145    2147    2149   2151   2152   2154   2222   10919  11789  12017
                      12022  12023  12053   12201   12270   12293  12301
$SS1  = 000000         2174#
$STUP = 177777         2134#
$SVLAD  045204        11799  11828#
$SVPC = 000200          789#   794
$SWR  = 167770          609#   619    651     652     653     654    655    656    657    658    906    907    908
                       2151   2152   2154    2155    2292    2351   2381   2542   2698   2730   2769   2830   3025
                       3158   3177   3209    3502    3675    3891   4127   4361   4524   4543   4568   4746   4765
                       4789   4947   5303    5635    5722    6064   6238   6600   6939   7305   7711   8124   8523
                       9163   9436   9927    10176   10449   10865  10914  10920  10935  10941  10943  11780  11781
                      11782  11783  11784   11790   11802   11804  11805  11808  11809  11810  11817  11818  11819
                      11830  11833  11836   12261   12262   12263  12264  12265  12278  12285  12290  12294  12302
                      12697
$SWRMK= 000000          658    659   11784   11785   11806
$TIMES  001212          906#  2151*  2292*   2351*   2381*   2542*  3209*  3502*  7305*  10865* 10920* 11817* 11824
```

```
                    11827* 11836
$TKB     001146       885# 11977  11998  12009  12034  12062  12089
$TKCNT   045714     11978# 11993* 12023  12040* 12154  12156*
$TKINT   045734      2182   2384   3029  11690  11993# 12014  12075
$TKQEN=  045733     11982# 12048  12159
$TKQIN   045716     11979# 11994* 11995  12046* 12047* 12048  12050*
$TKQOU   045720     11980# 11995* 12157  12158* 12159  12161*
$TKQSR   045722     11981# 11994  12050  12161
$TKS     001144       884# 11977  11999* 12030* 12032  12038* 12060  12076* 12086  12098* 12118*
$TKSRV   046004     11996  12009#
$TMP0    001176       900# 2448*  2449*
$TMP1    001200       901# 2307*  2746*  2747*  2785*  2786*  8571*  8930*  14257
$TMP2    001202       902#
$TMP3    001204       903#
$TMP4    001206       904#
$TMP5    001210       905#
$TN    = 000052       609#   619   2269  2281  2285  2292#  2293  2344  2351#  2353  2358  2367  2381#
                     2382   2523  2528  2542# 2544  2689  2698#  2701  2721  2730#  2731  2752  2756
                     2762   2769# 2770  2791  2802  2830#  2850  3015  3025#  3032  3145  3147  3154
                     3158#  3159  3169  3173  3177#  3178  3190  3209#  3218  3222  3245  3324  3492
                     3502#  3511  3519  3533  3536  3666  3675#  3883  3891#  4098  4127#  4145  4339
                     4361#  4520  4524#  4525  4539  4543#  4544  4558  4568#  4742  4746#  4747  4761
                     4765#  4766  4779  4789#  4925  4947#  4959  4962  5283  5303#  5304  5625  5628
                     5635#  5637  5691  5695  5722#  6052  6064#  6209  6238#  6571  6600#  6910  6939#
                     7273   7305#  7306  7318  7680  7711#  8089  8124#  8504  8523#  9150  9163#  9410
                     9436#  9906  9927#  10153  10176#  10188  10422  10449#  10461  10464  10853  10865#
$TPB     001152       887# 11963* 11974
$TPFLG   001157       891# 11921  11974
$TPS     001150       886# 11961  11974
$TRAP    050720      2147  12622#
$TRAP2   050742     12633# 12644
$TRP   = 000014     12637# 12646# 12647# 12648# 12649# 12650# 12651  12652# 12653  12654# 12655# 12656# 12657#
                    12658#
$TRPAD   050754     12627  12644#
$TSTNM   001102       864# 10881* 10919* 11779  11806  11828* 11833  11837  12277  12302
$TTYIN   046762     12171  12172  12189  12193#
$TYPBN=  ****** U   12650
$TYPDS   045250     11849# 12649
$TYPE    045474     11921# 12637  12645
$TYPEC   045644     11942  11949  11956  11961# 11962  12120
$TYPEX   045712     11967  11969  11972#
$TYPOC   050516     12567# 12646
$TYPON   050532     12566  12569# 12648
$TYPOS   050472     12562# 12647
$XTSTR   045010     11793#
$$GET4=  000000     10935#
$OFILL   050715     12563* 12567* 12577  12612#
$40CAT=  ****** U   11790  12287
.      = 072254      777#   781#   789   790#  792#  794#  798#  802#  854#  861#  912  2044#  2045#
                    2046#  2063#  2082#  2140  2154  2155  2196#  2201#  2205#  2209#  2218#  2315#  2330#
                    2337#  2418#  2422#  2487#  2493#  2499#  2580#  2618#  2663#  2669#  2675#  2709#  2713#
                    2719#  2862#  3553#  3571#  3581#  3650#  3654#  4985#  5252#  7456#  10541#  10547#  10808#
                    10877#  10943  10947  11538#  11553#  11706#  11730#  11742#  11754#  11836  11837  11903#  11974
                    11977  11981#  11982  11983#  12193#  12194  12201  12255  12302  12333#  12341#  12405#  12413#
                    12421#  12429#  12437#  12445#  12469#  12477#  12534#  12675  12699  14331#
```

```
LOAD     647#  3754  3756  3758  3759  3761  3763  3765  3767  3769  3771  3773  3795  3907  3909
        3911  3912  3914  3916  3918  3920  3922  3924  3926  3948  4184  4186  4188  4189  4191
        4193  4195  4197  4199  4201  4203  4225  4377  4379  4381  4382  4384  4386  4388  4390
        4392  4394  4396  4397  4581  4583  4585  4586  4588  4590  4592  4594  4596  4598  4600
        4622  4803  4805  4807  4808  4810  4812  4814  4816  4818  4820  4822  4823  5033  5037
        5041  5044  5048  5052  5056  5060  5064  5068  5072  5096  5332  5335  5338  5340  5343
        5346  5349  5355  5357  5359  5381
MAKECL   647#  3153  3172  4519  4537  4741  4759
MANUAL   647#  2839  3234  3522  4134  4948  10450
MSG     2284#  2287  2343#  2346  2366#  2369  2527#  2530  2688#  2691  2720#  2723  2761#  2764  2802#
        2804  3014#  3017  3189#  3192  3491#  3494  3665#  3668  3883#  3885  4097#  4100  4337#  4341
        4556#  4560  4777#  4781  4925#  4927  5283#  5285  5628#  5630  5695#  5697  6051#  6054  6208#
        6211  6571#  6573  6910#  6912  7273#  7275  7679#  7682  8088#  8091  8503#  8506  9149#  9152
        9410#  9412  9905#  9908  10152#  10155  10421#  10424  10853#  10855
MULT       1#  774#
NEWTST     1#  774#  2285  2344  2367  2528  2689  2721  2762  2802  3015  3154  3173  3190  3492
        3666  3883  4098  4339  4520  4539  4557  4742  4761  4778  4925  5283  5628  5695  6052
        6209  6571  6910  7273  7680  8089  8504  9150  9410  9906  10153  10422  10853
OFFST    647#  5649
POP        1#  774#  3791  3944  4221  4618  5090  5377  9542  10022  10283  10997  11024  11052  11077
       11199  11311  11377  11402  11433  11488  11586  11667  11890  12244  12684  12685
PUSH       1#  774#  3775  3928  4205  4602  5074  5361  9525  10005  10266  10961  11013  11042  11070
       11189  11252  11334  11389  11419  11456  11576  11639  11849  12218  12665  12671
REPORT     1#  774#
RFORGC   647#
RHCLEA   647#  2546  2733  2772  2832  2879  2928  3034  3224  3391  3538  3589  3677  3747  3893
        4129  4170  4363  4570  4791  4964  4992  5306  5319  5612  5639  5671  5724  5905  6066
        6240  6409  6602  6760  6941  7110  7308  7511  7713  7921  8126  8341  8525  8572  8744
        8940  9165  9308  9438  9478  9664  9782  9929  9969  10178  10230  10466  10585  10711
RH70CK   647#  7313  10183
SAVE     647#  12271
SAVTST   647#  2293  2353  2382  2698  2731  2770  3159  3178  4525  4544  4747  4766
SCH      647#
SCOPE    669#  2291  2350  2380  2541  2697  2729  2768  2829  3024  3157  3176  3208  3501  3674
        3890  4126  4360  4523  4542  4567  4745  4764  4788  4946  5302  5634  5670  5721  6063
        6237  6599  6938  7304  7710  8123  8522  9162  9435  9926  10175  10448  10864  10918
SEEKCO   647#  8946  9198  9450  9942  10202
SETPRI     1#  774#  12150
SETTRA 12637#  12646  12647  12648  12649  12651  12653  12654  12655  12656  12657
SETUP      1#  774#  2135
SKIP       1#  647#  774#  2269  2281  2358  2523  2701  2752  2756  2791  5625  5691
SLASH      1#  774#
SPACE    774#
SREGIS   647#  2944  3049  3256  3694  3809  4235  4407  4631  4831  5105  5388  5508  5790  5939
        6101  6295  6460  6647  6801  6996  7161  7374  7564  7806  7973  8220  8391  8638  8802
        8951  9203  9595  9690  9808  10056  10320  10531  10615  10796
STARS      1#  639  645  774#  787  853  857  912  1699  1701  1741  1743  1984  1989  2236
        2258  2285  2290  2344  2349  2367  2379  2458  2478  2528  2540  2591  2610  2637  2676
        2689  2696  2721  2728  2762  2767  2802  2828  3015  3023  3140  3150  3154  3156  3173
        3175  3190  3207  3211  3220  3492  3500  3504  3513  3666  3673  3883  3889  4098  4125
        4339  4359  4509  4515  4520  4522  4539  4541  4558  4566  4732  4738  4742  4744  4761
        4763  4779  4787  4925  4945  5283  5301  5495  5628  5633  5695  5720  6052  6062  6209
        6236  6571  6598  6910  6937  7273  7303  7680  7709  8089  8122  8504  8521  9150  9161
        9410  9434  9906  9925  10153  10174  10422  10447  10851  10852  10853  10863  10911  11759  11769
       11776  11839  11906  11976  12053  12068  12139  12163  12204  12257  12539  12616  12661  12677
START    647#  2544  2830  3032  3222  3536  3675  3891  4127  4361  4568  4789  4962  5304  5636
```

```
        5722    6064    6238    6600    6939    7306    7711    8124    8523    9163    9436    9927   10176   10464
SWRSU      1#    774#   2156#
TJUMP    647#    2850    3145    3147    3169    3218    3245    3324    3511    3519    3533    4145    4959    7318   10188
        10461
TRMTRP  12637#
TSCLR2    647#
TSCLR5    647#
TTSTNO    647#    2545    2831    3033    3223    3537    3676    3892    4128    4362    4569    4790    4963    5305    5638
        5723    6065    6239    6601    6940    7307    7712    8125    8524    9164    9437    9928   10177   10465
TYPBIN      1#    774#
TYPDEC      1#    774#   10930
TYPNAM      1#    774#
TYPNUM      1#    774#
TYPOCS      1#    774#
TYPOCT      1#    774#    2625    2631   12081
TYPTXT      1#    774#    2188    2193    2198    2202    2206    2211    2215    2270    2312    2327    2334    2388    2414
        2419    2423    2427    2484    2490    2496    2572    2577    2615    2621    2627    2659    2665    2671    2702
        2706    2710    2716    2853    2859    2885    2891    3274    3280    3300    3306    3550    3554    3567    3574
        3578    3605    3611    3647    3651    4148    4159    4982    4986    5177    5242    5249    7445    7452   10538
       10544   10804   10811   10868   10874   11522   11528   11535   11539   11543   11550   11554   11679   11686   11702
       11709   11715   11721   11727   11731   11735   11739   11745   11750   11761   12330   12338   12346   12354   12362
       12370   12378   12386   12394   12402   12410   12418   12426   12434   12442   12450   12458   12466   12474
VECSET    647#    2951    3056    3701    3816    3952    4242    4414    4637    4837    5112    5395    5515    5803    5952
        6114    6308    6473    6660    6814    7009    7174    7387    7577    7819    7986    8233    8404    8549    8651
        8815    8958    9108    9172    9210    9366    9454    9608    9703    9822    9946   10069   10207   10333   10504
       10622   10750
WT        647#    2967    3074    3481    3639    3717    3832    3968    4272    4444    4652    4852    5013    5275    5412
        5532    5660    5683    5822    5969    6131    6327    6490    6679    6831    7028    7191    7405    7593    7838
        8005    8252    8421    8564    8672    8832    8976    9123    9189    9228    9381    9471    9623    9718    9839
        9961   10086   10222   10351   10520   10637   10765
WTT       647#    2973    3080    3487    3645    3723    3838    3974    4279    4451    4658    4858    5019    5281    5418
        5538    5666    5689    5828    5975    6137    6333    6496    6685    6837    7034    7197    7411    7599    7844
        8011    8258    8427    8570    8678    8838    8982    9129    9195    9234    9387    9477    9629    9724    9845
        9967   10092   10228   10357   10526   10643   10771
$$CMRE    855#    894     895     896     897     898     899
$$CMTM    855#    900     901     902     903     904     905
$$ESCA      1#    774#
$$NEWT      1#    774#    2285    2344    2367    2528    2689    2721    2762    2802    3015    3154    3173    3190    3492
        3666    3883    4098    4339    4520    4539    4558    4742    4761    4779    4925    5283    5628    5695    6052
        6209    6571    6910    7273    7680    8089    8504    9150    9410    9906   10153   10422   10853
$$SET   12637#   12646   12647   12648   12649   12651   12653   12654   12655   12656   12657
$$SKIP      1#    774#
.EQUAT      1#    609#     664
.HEADE      1#    609#
.KT11       1#    609#     818
.SETUP      1#    609#    2133
.SWRHI      1#    609#     647
.SWRLO    609#    659#     660
.$ACT1      1#    609#     785
.$APTB      1#
.$APTH      1#
.$APTY      1#
.$ASTA      1#
.$CATC      1#    609#     775
.$CMTA      1#    609#     855
.$DB2D      1#
```

```
.$DB2O      1#
.$DIV       1#
.$EOP       1#    609#  10909
.$ERRO      1#    609#  12255
.$ERRT      1#    609#
.$MULT      1#
.$POWE      1#    609#  12659
.$RAND      1#
.$RDDE      1#
.$RDOC      1#    609#  12202
.$READ      1#    609#  11974
.$R2AZ      1#
.$SAVE      1#
.$SB2D      1#
.$SB2O      1#
.$SCOP      1#    609#  11774
.$SIZE      1#
.$SUPR      1#
.$TRAP      1#    609#  12614
.$TYPB      1#
.$TYPD      1#    609#  11837
.$TYPE      1#    609#  11904
.$TYPO      1#    609#  12537
.$40CA      1#
.1170       1#


. ABS.  072254     000

 ERRORS DETECTED:  0

DSKZ:CZRJ1D.BIN,DSKZ:CZRJ1D.LST/CRF/SOL/NL:TOC:MD:MC:CND/LI:ME=CZRJ1D.SML,CZRJ1D.P11
RUN-TIME: 110 161 8 SECONDS
RUN-TIME RATIO: 569/281=2.0
CORE USED:  38K  (75 PAGES)
```