RP04/5/6

RP04/5/6 FCTNL 2
CZRJJD0

AH 9225D MC

FICHE 1 OF 2

NOV 1979
3030040
MADE IN USA

RP04/5/6

RP04/5/6 FCTNL 2

CZRJJD0

AH-9225D-MC

FICHE 2 OF 2

NOV 1979

digital

MADE IN USA

.REM @

IDENTIFICATION

PRODUCT CODE: AC-9223D-MC

PRODUCT NAME: CZRJJDO RP04/5/6 FUNCTIONAL CONTROLLER TEST PART II

DATE CREATED: MAY, 1979

MAINTAINER: DIAGNOSTIC ENGINEERING

AUTHOR: PETE BLACKSTONE

DEC DECUS DECTAPE

CONTENTS

1.0 ABSTRACT

THIS DIAGNOSTIC TESTS THE DCL OF THE RP04/5/6 DISK SUBSYSTEM WHEN CONNECTED TO EITHER 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 STATIC 1 (DZRPS AND DZRPT) HAS 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.

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

AND IT ASSUMES THAT MAINDEC-11-DZRJI-(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 DERSIRES. 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 210---FOR UNIT SELECTION

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

204 RESTART
SAME AS 200 START, WITH THE FOLLOWING EXCEPTIONS: THE
PROGRAM WILL INTERROGATE THE OPERATOR FOR THE NON-DEFAULT
C.S.R. AND VECTOR ADDRESS FOR THE RHXX CONTROLLER.
WHEN THESE QUESTIONS HAVE CORRECTLY BEEN ANSWERED, THE
PROGRAM WILL AUTOMATICALLY RESTART FROM ADDRESS 200.

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

4.3 PROGRAM AND/OR OPERATOR ACTION

1. LOAD THE PROGRAM INTO MEMORY.

2. SET STARTING ADDRESS ON THE SWITCH REGISTER

3. PRESS "LOAD ADDRESS".

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

5. PRESS "START".

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

5.0 OPERATING PROCEDURE

5.1 OPERATIONAL SWITCH SETTINGS

IF THE PROGRAM IS BEING RUN ON A SWITCHLESS PROCESSOR (I. E.
AN 11/34) IT WILL DETERMINE THAT A HARDWARE SWITCH REGISTER
IS NOT PRESENT, AND WILL USE A "SOFTWARE" SWITCH REGISTER.
THE SETTINGS OF THE "SOFTWARE" SWITCHES ARE CONTROLLED
THROUGH A KEYBOARD ROUTINE WHICH IS CALED BY TYPING A
'CONTROL G'. THE PROGRAM WILL RECOGNIZE A 'CONTROL G' AT ANY
TIME EXCEPT WHEN IT IS AT A HIGHER PRIORITY PROCESSING AN
RP04/5/6 INTERRUPT. THE "SOFTWARE" SWITCH VALUES 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 REGISTERS, THE #SOFTWARE"
SWITCH REGISTER MAY ALSO BE USED. IF THE PROGRAM FINDS ALL
16 SWITCHES IN THE 'UP' POSITION WHEN IT IS STARTED, ALL
SWITCH REGISTER REFERENCES WILL BE TO THE "SOFTWARE" REGISTER
AND THE PROCEDURES DESCRIBED ABOVE MUST BE FOLLOWED.

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

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

SWITCH 14 - LOOP ON TEST
WHEN THIS SWITCH IS SET THE PROGRAM WILL BEGIN TO LOOP

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

SWITCH 13 - INHIBIT ERROR TYPEOUTS
WHEN THIS SWITCH IS SET FURTHER ERROR PRINTOUTS WILL
CEASE, HOWEVER OPERATOR INSTRUCTIONS SUCH AS "STOP DRIVE X"
WILL CONTINUE. AT THE END OF PASS "TOTAL NUMBER OF ERRORS
ON THIS PASS ON DRIVE X" WILL BE TRUE, THAT IS, ALTHOUGH
PRINTOUTS WERE INHIBITED IF THAT PASS FOUND 6 ERRORS,
IT WILL SAY SO.

SWITCH 12 - RH70 CONTROLLER SELECT
THIS SWITCH MUST BE SET AT THE START OF THE PROGRAM WHEN THE
DISK DRIVES TO 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 STATED
ADDRESS TO DETERMINE THE CAUSE OF THE ERROR.

6.1 'FATAL' ERRORS

IN THE EVENT THAT THE DISK DRIVE ECOMES UNAVAILABLE TO THE
CONTROLLER, POWERS DOWN, OR CERTAIN CRITICAL STATUS BITS
CANOT BE CLEARED PRIOR TO THE START OF ATEST SEQUENCE - THIS
INFORMATION WILL BE COMMUNICATED TO THE OPERATOR. IN
ADDITION, THE TTY BELL WILL RING AND THE PROGRAM WILL HALT.
IT IS SUGGESTED THAT IF THIS HAPPENS, THE OPERATOR LOAD
ADDRESS 200 (210) AND RESTART THE PROGRAM AS A FIRST ATTEMPT
TO SOLVE THE PROBLEM. IF THE FILURE CONTINUES TO OCCUR,
THERE ARE TWO OPTIONS FOR 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 THE DZRPS DIAGNOSTIC 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 FOLLOWING 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 REGISTER FEATURE WHILE RUNNING 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 20 SECONDS. SUBSEQUENT PASSES WILL TAKE 60 SECONDS .

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  CZRJJDO, RP04/5/6 FCTNL CTRLR2
                        ;*COPYRIGHT (C) 1976,1978
                        ;*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
```

```
                        .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 CONTAIN 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  033104                  $ENDAD                  ;;1)SET LOC.46 TO ADDRESS OF $ENDAD IN .$EOP
                        000052          .=52
000052  020000                  .WORD   20000           ;;2)SET LOC.52 TO 20000
                        000200          .=$SVPC                 ;; RESTORE PC

                                .SBTTL  STARTING ADDRESS

                        000200          .=200
000200  000137  004712  RA:     JMP     @#BEGIN         ;NORMAL START
000204  000137  035460  ADDMOD: JMP     @#BASECH        ;GET DEVICE PARAMETERS
000210  000137  004676          JMP     @#BEGIN2        ;JUMP TO SELECT DRIVE START
                        000220          .=220
000220  000137  004662          JMP     @#BEGIN1        ;JUMP TO NO OPERATOR TESTS START

                                ;*STARTING ADDRESS 200 FOR NORMAL STARTS
                                ;*THIS WILL TEST ALL RP04'S ON THE SYSTEM A SINGLE DRIVE AT A TIME
                                ;*
                                ;*STARTING ADDRESS 210 WILL TEST ONLY ONE SPECIFIED DRIVE
                                ;*
                                ;*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
```

```
   904  001224  000012                $LF:    .ASCIZ  <12>            ;;LINE FEED
   905                                ;;*************************************************************
```

```
                        .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  043014                  EM1                     ;RP04 DID NOT INTERRUPT
                                                        ;WAITED ON BIT DID NOT OCCUR
001230  057737                  DH1                     ;PC
                                                        ;WAT PC
                                                        ;BIT WAITED
                                                        ;REG ADDRESS
                                                        ;REG CONTENTS
                                                        ;RHCS1 CONTENTS
001232  062210                  DT1                     ;$ERRPC,WAITPC,WAITBT,WAITRE,$BDDAT,CS1
001234  062530                  DF1                     ;0,0,0,0,0,0

                        ;ITEM2
001236  043043                  EM2                     ;INTERRUPT ENABLE BIT DOWN BUT
                                                        ;WAITED ON BIT DID NOT OCCUR
001240  057737                  DH1                     ;PC
                                                        ;WAT PC
                                                        ;BIT WAITED
                                                        ;REG ADDRESS
                                                        ;REG CONTENTS
                                                        ;RHCS1 CONTENTS
001242  062210                  DT1                     ;$ERRPC,WAITPC,WAITBT,WAITRE,$BDDAT,CS1
001244  062530                  DF1                     ;0,0,0,0,0,0

                        ;ITEM3
001246  043132                  EM3                     ;RP04 DID NOT INTERRUPT WHEN
                                                        ;WAITED ON BIT DID SET
001250  057737                  DH1                     ;PC
                                                        ;WAT PC
                                                        ;BIT WAITED
                                                        ;REG ADDRESS
                                                        ;RHCS1 CONTENTS
001252  062210                  DT1                     ;$ERRPC,WAITPC,WAITBT,WAITRE,$BDDAT,CS1
001254  062530                  DF1                     ;0,0,0,0,0,0

                        ;ITEM4
001256  043213                  EM4                     ;WAITED ON BIT DID SET BUT
                                                        ;TIME IS IN ERROR
```

```
                                                        ;TIME IS GIVEN IN 10 MICRO SEC.
                                                        ;(DECIMAL)
001260  060117                    DH4                 ;PC
                                                        ;WAT PC
                                                        ;BIT WAITED
                                                        ;REG ADDRESS
                                                        ;TIME IN 10 MSEC
001262  062230                    DT4                 ;$ERRPC,WAITPC,WAITBT,WAITRE,$BDDAT,WAITIM
001264  062537                    DF4                 ;0,0,0,0,0,1

                        ;ITEM5
001266  043324                    EM5                 ;RHAS DOES NOT CLEAR BY
                                                        ;MOVING IN ALL ONES
001270  060260                    DH5                 ;PC
                                                        ;REG. ADDR.
                                                        ;GOOD DATA
                                                        ;RECEIVED DATA
001272  062252                    DT5                 ;$ERRPC,REGADR,$GDDAT,$BDDAT
001274  062546                    DF5                 ;0,0,0,0

                        ;ITEM6
001276  043376                    EM6                 ;LOADING RHER1 FOR ALL
                                                        ;UNITS DID NOT SET ANY BITS
                                                        ;IN RHAS-NO UNITS PRESENT
001300  060377                    DH6                 ;PC
                                                        ;REG ADDR
                                                        ;RECEIVED DATA
001302  062266                    DT6                 ;$ERRPC,REGADR,$BDDAT
001304  062553                    DF6                 ;0,0,0

                        ;ITEM7
001306  043464                    EM7                 ;SPECIFIED REGISTER NONEXISTANT
                                                        ;SO ABORT PROGRAM
001310  060476                    DH7                 ;PC
                                                        ;ADDR. OF REG.
001312  062300                    DT7                 ;$ERRPC,TEMP1
001314  062557                    DF7                 ;0,0

                        ;ITEM10
001316  043534                    EM10                ;STOPED DRIVE HAS MOL BIT
                                                        ;IN RHDS1 = 1
001320  060536                    DH10                ;PC
                                                        ;TEST NO
                                                        ;FAILING REG ADDR
                                                        ;CONTENTS OF RHCS1
                                                        ;CONTENTS OF RHCS2
                                                        ;CONTENTS OF RHDS1
                                                        ;CONTENTS OF RHER1
001322  062310                    DT10                ;$ERRPC,$TSTNM,$BDADR,CS1,CS2,DS1,ER1
001324  062562                    DF10                ;0,0,0,0,0,0,0

                        ;ITEM11
001326  043603                    EM11                ;WITH SPINDLE POWERED DOWN
                                                        ;RHCS2 SHOULD HAVE ONLY
                                                        ;UNIT NUMBER AND IR HIGH
001330  060536                    DH10                ;PC
```

```
1018                                                    ;TEST NO
1019                                                    ;FAILING REG. ADR
1020                                                    ;CONTENTS OF RHCS1
1021                                                    ;CONTENTS OF RHCS2
1022                                                    ;CONTENTS OF RHDS1
1023                                                    ;CONTENTS OF RHER1
1024  001332  062310                    DT10            ;$ERRPC,$TSTNM,$BDADR,CS1,CS2,DS1,ER1
1025  001334  062562                    DF10            ;0,0,0,0,0,0,0
1026
1027                              ;ITEM12
1028  001336  043710                    EM12            ;AFTER A POWER UP WITH
1029                                                    ;NO PACK ACKNOWLEDGE COMMAND
1030                                                    ;RHDS1 SHOULD HAVE MOL=1, VV=0
1031  001340  060536                    DH10            ;PC
1032                                                    ;TEST NO
1033                                                    ;FAILING REGISTER ADDR.
1034                                                    ;CONTENTS OF RHCS1
1035                                                    ;CONTENTS OF RHCS2
1036                                                    ;CONTETNS OF RHDS1
1037                                                    ;CONTENTS OF RHER1
1038  001342  062310                    DT10            ;$ERRPC,$TSTNM,$BDADR,CS1,CS2,DS1,ER1
1039  001344  062562                    DF10            ;0,0,0,0,0,0,0
1040
1041                              ;ITEM13
1042  001346  044016                    EM13            ;AFTER A POWER UP WITHOUT
1043                                                    ;ANY INIT RHCS1 SHOULD
1044                                                    ;HAVE GO=0, DVA=1, RDY=1
1045                                                    ;IE=0, DISREGARD
1046                                                    ;ALL OTHER BITS
1047  001350  060536                    DH10            ;PC
1048                                                    ;TEST NO
1049                                                    ;FAILING REGISTER ADDR.
1050                                                    ;CONTENTS OF RHCS1
1051                                                    ;CONTENTS OF RHCS2
1052                                                    ;CONTENTS OF RHDS1
1053                                                    ;CONTENTS OF RHER1
1054  001352  062310                    DT10            ;$ERRPC,$TSTNM,$BDADR,CS1,CS2,DS1,ER1
1055  001354  062562                    DF10            ;0,0,0,0,0,0,0
1056
1057                              ;ITEM14
1058  001356  044135                    EM14            ;AFTER POWER UP RHCC
1059                                                    ;SHOULD BE=0
1060  001360  060260                    DH5             ;PC
1061                                                    ;REG. ADDR.
1062                                                    ;GOOD DATA
1063                                                    ;RECEIVED DATA
1064  001362  062252                    DT5             ;$ERRPC,REGADR,$GDDAT,$BDDAT
1065  001364  062546                    DF5             ;0,0,0,0
1066
```

```
1067                                    ;ITEM15
1068  001366  044207                            EM15                    ;PACK ACKNOWLEDGE CAUSED
1069                                                                    ;AN ERROR
1070                                                                    ;GOOD DATA IS BEFORE COMMAND
1071                                                                    ;RECEIVED DATA IS AFTER COMMAND
1072  001370  060260                            DH5                     ;PC
1073                                                                    ;REG. ADDR.
1074                                                                    ;GOOD DATA
1075                                                                    ;RECEIVED DATA
1076  001372  062252                            DT5                     ;$ERRPC,REGADR,$GDDAT,$BDDAT
1077  001374  062546                            DF5                     ;0,0,0,0
1078
1079                                    ;ITEM16
1080  001376  044350                            EM16                    ;GIVING A NO-OP COMMAND CAUSED
1081                                                                    ;AN ERROR
1082                                                                    ;GOOD DATA GIVES REGISTER
1083                                                                    ;CONTENTS BEFORE COMMAND
1084                                                                    ;RECEIVED DATA GIVES REGISTER
1085                                                                    ;CONTENTS AFTER COMMAND
1086  001400  060260                            DH5                     ;PC
1087                                                                    ;REG. ADDR.
1088                                                                    ;GOOD DATA
1089                                                                    ;RECEIVED DATA
1090  001402  062252                            DT5                     ;$ERRPC,REGADR,$GDDAT,$BDDAT
1091  001404  062546                            DF5                     ;0,0,0,0
1092
1093                                    ;ITEM17
1094  001406  044476                            EM17                    ;DRIVE CLEAR COMMAND
1095                                                                    ;CAUSED AN ERROR
1096                                                                    ;GOOD DATA GIVES WHAT SHOULD
1097                                                                    ;BE THERE
1098                                                                    ;RECEIVED DATA GIVES WHAT WAS
1099                                                                    ;THERE AFTER COMMAND
1100  001410  060260                            DH5                     ;PC
1101                                                                    ;REG. ADDR.
1102                                                                    ;GOOD DATA
1103                                                                    ;RECEIVED DATA
1104  001412  062252                            DT5                     ;$ERRPC,REGADR,$GDDAT,$BDDAT
1105  001414  062546                            DF5                     ;0,0,0,0
1106
1107                                    ;ITEM20
1108  001416  044633                            EM20                    ;READ-IN COMMAND GAVE AN ERROR
1109                                                                    ;GOOD DATA HAS WHAT SHOULD BE THERE
1110                                                                    ;RECEIVED DATA HAS WHAT WAS
1111                                                                    ;AFTER COMMAND
1112  001420  060260                            DH5                     ;PC
1113                                                                    ;REG. ADDR.
1114                                                                    ;GOOD DATA
1115                                                                    ;RECEIVED DATA
1116  001422  062252                            DT5                     ;$ERRPC,REGADR,$GDDAT,$BDDAT
1117  001424  062546                            DF5                     ;0,0,0,0
1118
1119
1120                                    ;ITEM 21
1121  001426  045002                            EM21                    ;RHCS1 CONTENTS DURING
1122                                                                    ;COMMAND WAS IN ERROR
```

```
1123  001430  060260                          DH5
1124  001432  062252                          DT5
1125  001434  062546                          DF5
1126
1127                                  ;ITEM 22
1128  001436  045055                          EM22            ;RHDS1 CONTENTS DURING
1129                                                          ;COMM ANS WAS IN ERROR
1130  001440  060260                          DH5
1131  001442  062252                          DT5
1132  001444  062546                          DF5
1133
1134                                  ;ITEM 23
1135  001446  045130                          EM23            ;UNLOAD COMMAND GAVE AN ERROR
1136                                                          ;GOOD DATA GIVES WHAT SHOULD
1137                                                          ;BE THERE
1138                                                          ;RECEIVED DATA GIVES WHAT WAS
1139                                                          ;THERE AFTER COMMAND
1140  001450  060260                          DH5
1141  001452  062252                          DT5
1142  001454  062546                          DF5
1143
1144                                  ;ITEM 24
1145  001456  045277                          EM24            ;OFFSET COMMAND CAUSED AN ERROR
1146                                                          ;GOOD DATA IS WHAT SHOULD BE THERE
1147                                                          ;RECEIVED DATA GIVES WHAT WAS THERE
1148                                                          ;AFTER AN OFFSET COMMAND
1149  001460  060260                          DH5
1150  001462  062252                          DT5
1151  001464  062546                          DF5
1152
1153                                  ;ITEM 25
1154  001466  045442                          EM25            ;RETURN TO CENTER LINE COMMAND
1155                                                          ;CAUSED AN ERROR
1156                                                          ;GOOD DATA GIVES WHAT SHOULD BE
1157                                                          ;THERE
1158                                                          ;RECEIVED DATA GIVES WHAT WAS
1159                                                          ;THERE AFTER COMMAND
1160  001470  060260                          DH5
1161  001472  062252                          DT5
1162  001474  062546                          DF5
1163
1164                                  ;ITEM 26
1165  001476  045624                          EM26            ;500 OFFSETS CAUSED AN ERROR
1166  001500  060715                          DH26            ;PC
1167                                                          ;CONT. OF RHCS1
1168                                                          ;CONT. OF RHCS2
1169                                                          ;CONT. OF RHDS1
1170                                                          ;CONT. OF RHER1
1171                                                          ;CONT. OF RHER2
1172                                                          ;CONT. OF RHER3
1173  001502  062330                          DT26            ;$ERRPC,CS1,CS2,DS1,ER1,ER2,ER3
1174  001504  062571                          DF26            ;0,0,0,0,0,0,0
1175
1176                                  ;ITEM 27
1177  001506  045714                          EM27            ;WRITE HEADER AND DATA
1178                                                          ;CAUSED IMPROPER REGISTER CHANGE
```

```
1179                                                    ;GOOD DATA GIVES WHAT
1180                                                    ;SHOULD BE THERE
1181                                                    ;RECEIVED DATA GIVES WHAT
1182                                                    ;WAS THERE AFTER COMMAND
1183  001510  060260                    DH5
1184  001512  062252                    DT5
1185  001514  062546                    DF5
1186
1187                              ;ITEM 30
1188  001516  046132                    EM30           ;WRITE HEADER AND DATA
1189                                                    ;CHANGED WRITE FROM BUFFER
1190  001520  061115                    DH30           ;PC
1191                                                    ;WORD NO
1192                                                    ;GOOD DATA
1193                                                    ;BAD DATA
1194  001522  062352                    DT30           ;$ERRPC,ERWORD,$GDDAT,$BDDAT
1195  001524  062601                    DF30           ;0,0,0,0
1196
1197                              ;ITEM 31
1198  001526  046212                    EM31           ;READ HEADER AND DATA CAUSED
1199                                                    ;IMPROPER REGISTER CHANGE
1200                                                    ;GOOD DATA HAS WHAT SHOULD
1201                                                    ;BE THERE
1202                                                    ;RECEIVED DATA GIVES WHAT
1203                                                    ;WAS THERE AFTER COMMAND
1204  001530  060260                    DH5
1205  001532  062252                    DT5
1206  001534  062546                    DF5
1207
1208                              ;ITEM 32
1209  001536  046427                    EM32           ;WRITE HEADER AND DATA FOLLOWED
1210                                                    ;BY A READ HEADER AND DATA
1211                                                    ;CAUSED A READ/WRITE ERROR
1212  001540  061115                    DH30
1213  001542  062352                    DT30
1214  001544  062601                    DF30
1215
1216                              ;ITEM 33
1217  001546  046534                    EM33           ;READ DATA CAUSED IMPROPER REGISTER
1218                                                    ;CHANGE
1219                                                    ;GOOD DATA GIVES WHAT SHOULD BE THERE
1220                                                    ;RECEIVED DATA GIVES WHAT WAS THERE AFTER
1221                                                    ;COMMAND
1222  001550  060260                    DH5
1223  001552  062252                    DT5
1224  001554  062546                    DF5
1225
1226                              ;ITEM 34
1227  001556  046736                    EM34           ;READ DATA INCORRECT
1228  001560  061115                    DH30
1229  001562  062352                    DT30
1230  001564  062601                    DF30
1231
1232                              ;ITEM 35
1233  001566  046762                    EM35           ;WRITE DATA COMMAND CAUSED
1234                                                    ;IMPROPER REGISTER CHANGE
```

```
1235                                                    ;GOOD DATA GIVES WHAT SHOULD BE THERE
1236                                                    ;RECEIVED DATA GIVES REGISTER
1237                                                    ;CONTENTS AFTER WRITE DATA
1238  001570  060260                    DH5
1239  001572  062252                    DT5
1240  001574  062546                    DF5
1241
1242                              ;ITEM 36
1243  001576  047200                    EM36           ;WRITE DATA COMMAND CHANGED
1244                                                    ;WRITE FROM BUFFER
1245  001600  061115                    DH30
1246  001602  062352                    DT30
1247  001604  062601                    DF30
1248
1249                              ;ITEM 37
1250  001606  047255                    EM37           ;SEEK COMMAND CAUSED AN
1251                                                    ;ERROR
1252                                                    ;GOOD DATA GIVES WHAT SHOULD
1253                                                    ;BE THERE
1254                                                    ;RECEIVED DATA GIVES WHAT
1255                                                    ;WAS THERE AFTER SEEK COMMAND
1256  001610  060260                    DH5            ;
1257  001612  062252                    DT5            ;
1258  001614  062546                    DF5            ;
1259
1260                              ;ITEM 40
1261  001616  047472                    EM40           ;WRITE CHECK CAUSED AN
1262                                                    ;IMPROPER REGISTER CHANGE
1263                                                    ;GOOD DATA GIVES WHAT SHOULD
1264                                                    ;BE THERE
1265                                                    ;RECEIVED DATA GIVES WHAT WAS
1266                                                    ;THERE AFTER COMMAND
1267  001620  060260                    DH5
1268  001622  062252                    DT5
1269  001624  062546                    DF5
1270
1271                              ;ITEM 41
1272  001626  047701                    EM41           ;LOCKING OUT WRITES BY WRITE
1273                                                    ;LOCK BUTTON CAUSED IMPROPER
1274                                                    ;REGISTER CHANGE
1275                                                    ;GOOD DATA GIVES WHAT SHOULD
1276                                                    ;BE THERE
1277                                                    ;RECEIVED DATA GIVES WHAT
1278                                                    ;WAS THERE AFTER WRITES
1279                                                    ;WERE LOCKED OUT BY
1280                                                    ;BUTTON
1281  001630  060260                    DH5
1282  001632  062252                    DT5
1283  001634  062546                    DF5
1284
1285                              ;ITEM 42
1286  001636  050162                    EM42           ;ATTEMPTING TO WRITE WITH WRITE
1287                                                    ;LOCKED OUT CAUSED IMPROPER
1288                                                    ;REGISTER CHANGE
1289                                                    ;GOOD DATA GIVES WHAT SHOULD
1290                                                    ;BE THERE
```

```
1291                                                    ;RECEIVED DATA GIVES WHAT WAS
1292                                                    ;THERE AFTER ATTEMPT
1293  001640  060260                    DH5
1294  001642  062252                    DT5
1295  001644  062546                    DF5
1296
1297                              ;ITEM 43
1298  001646  050440                    EM43           ;WRITING WITH WRITE LOCKED
1299                                                    ;OUT CHANGED DISK DATA
1300                                                    ;GOOD DATA GIVES WHAT WAS
1301                                                    ;ON DISK BEFORE WRITE WITH
1302                                                    ;WRITE LOCK WAS ATTEMPTED
1303                                                    ;RECEIVED DATA GIVES WHAT WAS
1304                                                    ;READ BACK AFTER WRITE WITH
1305                                                    ;WRITE LOCK WAS ATTEMPTED
1306  001650  061115                    DH30
1307  001652  062352                    DT30
1308  001654  062601                    DF30
1309
1310                              ;ITEM 44
1311  001656  050776                    EM44           ;ENABLING WRITES BY WRITE LOCK
1312                                                    ;BUTTON CAUSED AN ERROR
1313                                                    ;GOOD DATA GIVES WHAT SHOULD
1314                                                    ;BE THERE
1315                                                    ;RECEIVED DATA GIVES WHAT WAS
1316                                                    ;THERE AFTER WRITE LOCK
1317                                                    ;BUTTON ENABLED WRITES
1318  001660  060260                    DH5            ;
1319  001662  062252                    DT5            ;
1320  001664  062546                    DF5            ;
1321
1322                              ;ITEM 45
1323  001666  051270                    EM45           ;TRANSFERRING ON LAST BLOCK IE. CYLINDER
1324                                                    ;410, SECTOR 21, TRACK 18
1325                                                    ;CAUSED IMPROPER REGISTER
1326                                                    ;CHANGE
1327                                                    ;GOOD DATA GIVES WHAT SHOULD
1328                                                    ;BE THERE
1329                                                    ;RECEIVED DATA GIVES WHAT WAS
1330                                                    ;THERE AFTER TRANSFER
1331  001670  060260                    DH5
1332  001672  062252                    DT5
1333  001674  062546                    DF5
1334
1335                              ;ITEM 46
1336  001676  051576                    EM46           ;DATA READ FROM LAST
1337                                                    ;BLOCK IE. CYLINDER 410
1338                                                    ;SECTOR 21, TRACK 18 IS IN
1339                                                    ;ERROR
1340  001700  061115                    DH30
1341  001702  062352                    DT30
1342  001704  062601                    DF30
1343
1344                              ;ITEM 47
1345  001706  051722                    EM47           ;TRANSFERRING FROM NONEXISTANT
1346                                                    ;SECTOR CAUSED IMPROPER
```

```
1347                                                        ;REGISTER CHANGE
1348                                                        ;GOOD DATA GIVES WHAT SHOULD
1349                                                        ;BE THERE
1350                                                        ;RECEIVED DATA GIVES WHAT WAS
1351                                                        ;THERE AFTER ATTEMPTED
1352                                                        ;TRANSFER
1353  001710  060260                    DH5
1354  001712  062252                    DT5
1355  001714  062546                    DF5
1356
1357                              ;ITEM 50
1358  001716  052204                    EM50               ;TRANSFERRING FROM NONEXISTANT
1359                                                        ;SECTOR CAUSED DATA ERROR
1360                                                        ;GOOD DATA GIVES WHAT
1361                                                        ;SHOULD BE IN BUFFER
1362                                                        ;RECEIVED DATA GIVES WHAT WAS,
1363                                                        ;IN BUFFER AFTER TRANSFER
1364  001720  061115                    DH30
1365  001722  062352                    DT30
1366  001724  062601                    DF30
1367
1368                              ;ITEM 51
1369  001726  052423                    EM51               ;GIVING ILLEGAL FUNCTION CAUSED
1370                                                        ;IMPROPER REGISTER CHANGE
1371                                                        ;GOOD DATA GIVES WHAT SHOULD BE
1372                                                        ;THERE
1373                                                        ;RECEIVED DATA GIVES REGISTER
1374                                                        ;CONTENTS AFTER ILLEGAL FUNCTION
1375  001730  061227                    DH51               ;PC
1376                                                        ;REG. ADDR.
1377                                                        ;GOOD DATA
1378                                                        ;RECEIVED DATA
1379                                                        ;ILLEGAL FUNCTION
1380  001732  062366                      DT51             ;$ERRPC,REGADR,$GDDAT,$BDDAT,ILLEGL
1381  001734  062606                      DF51             ;0,0,0,0,0
1382
1383
1384                              ;ITEM 52
1385  001736  052670                    EM52               ;WRITE DATA ON NONEXISTANT
1386                                                        ;SECTOR CAUSED IMPROPER
1387                                                        ;REGISTER CHANGE
1388                                                        ;GOOD DATA GIVES WHAT SHOULD
1389                                                        ;BE THERE
1390                                                        ;RECEIVED DATA GIVES WHAT
1391                                                        ;WAS THERE AFTER ATTEMPTED
1392                                                        ;WRITE DATA
1393  001740  060260                    DH5
1394  001742  062252                    DT5
1395  001744  062546                    DF5
1396
1397                              ;ITEM 53
1398  001746  053141                    EM53               ;READ HEADER AND DATA AFTER
1399                                                        ;A SEARCH CAUSED AN ERROR
1400  001750  061115                    DH30
1401  001752  062352                    DT30
1402  001754  062601                    DF30
```

```
1403
1404                                    ;ITEM 54
1405  001756  053227                            EM54            ;ATTEMPTED OPERATION WITH
1406                                                            ;INVALID ADDRESS CAUSED
1407                                                            ;IMPROPER REGISTER CHANGE
1408                                                            ;GOOD DATA GIVES WHAT SHOULD
1409                                                            ;BE THERE
1410                                                            ;RECEIVED DATA GIVES WHAT WAS
1411                                                            ;THERE AFTER OPERATION
1412  001760  060260                                DH5
1413  001762  062252                                DT5
1414  001764  062546                                DF5
1415
1416                                    ;ITEM 55
1417  001766  053474                            EM55            ;WRITING/READING WITH EXPECTED
1418                                                            ;ADDRESS OVERFLOW ERROR CAUSED
1419                                                            ;IMPROPER REGISTER CHANGE
1420                                                            ;GOOD DATA GIVES WHAT SHOULD
1421                                                            ;BE THERE
1422                                                            ;RECEIVED DATA GIVES WHAT
1423                                                            ;WAS THERE AFTER OPERATION
1424  001770  060260                                DH5
1425  001772  062252                                DT5
1426  001774  062546                                DF5
1427
1428                                    ;ITEM 56
1429  001776  053762                            EM56            ;DATA READ WITH AN EXPECTED
1430                                                            ;ADDRESS OVERFLOW ERROR IS
1431                                                            ;INCORRECT
1432                                                            ;WORD NO 1 TO 260 SHOULD
1433                                                            ;BE READ
1434                                                            ;WORD NOS 261 TO 266 SHOULD
1435                                                            ;NOT CHANGE DUE TO READ
1436  002000  061115                                DH30
1437  002002  062352                                DT30
1438  002004  062601                                DF30
1439
1440                                    ;ITEM 57
1441  002006  054172                            EM57            ;ATTEMPTING DATA COMMAND
1442                                                            ;WITH WRONG FORMAT BIT CAUSED
1443                                                            ;IMPROPER REGISTER CHANGE
1444                                                            ;GOOD DATA GIVES WHAT SHOULD BE
1445                                                            ;THERE
1446                                                            ;RECEIVED DATA GIVES WHAT WAS
1447                                                            ;THERE AFTER ATTEMPTED DATA
1448                                                            ;TRANSFER
1449  002010  060260                                DH5
1450  002012  062252                                DT5
1451  002014  062546                                DF5
1452
1453                                    ;ITEM 60
1454  002016  054464                            EM60            ;ATTEMPTING TO MODIFY REGISTER
1455                                                            ;DURING AN OPERATION CAUSED
1456                                                            ;IMPROPER REGISTER CHANGE
1457                                                            ;GOOD DATA GIVES WHAT SHOULD
1458                                                            ;BE THERE
```

```
1459                                                        ;RECEIVED DATA GIVES WHAT WAS
1460                                                        ;THERE AFTER OPERATION
1461                                                        ;WAS COMPLETE
1462  002020  061366                    DH60               ;PC
1463                                                        ;REG. ADDR.
1464                                                        ;GOOD DATA
1465                                                        ;RECEIVED DATA
1466                                                        ;MODFING REGISTER
1467  002022  062404                    DT60               ;$ERRPC,REGADR,$GDDAT,$BDDAT,$BDADR
1468  002024  062614                    DF60               ;0,0,0,0,0
1469
1470                              ;ITEM 61
1471  002026  055073                    EM61               ;DEVICE NOT AVAIBLE BEFOR COMMAND WAS TO BE GIVEN
1472  002030  061523                    DH61               ;PC
1473                                                        ;PC OF JSR
1474                                                        ;RHDS1
1475  002032  062422                    DT61               ;$ERRPC,PCJSR,$BDADR
1476  002034  062622                    DF61               ;0,0,0
1477
1478                              ;ITEM 62
1479  002036  055073                    EM61               ;DEVICE NOT AVAIBLE BEFOR COMMAND WAS TO BE GIVEN
1480  002040  061615                    DH62               ;PC
1481                                                        ;PC OF JSR
1482                                                        ;RHCS1 WAS
1483  002042  062434                    DT62               ;$ERRPC,PCJSR,$BDADR
1484  002044  062626                    DF62               ;0,0,0
1485
1486
1487                              ;ITEM 63
1488  002046  055157                    EM63               ;RHDS1 CONTENTS DURING
1489                                                        ;COMMAND WAS IN ERROR
1490  002050  060260                    DH5
1491  002052  062252                    DT5
1492  002054  062546                    DF5
1493
1494
1495                              ;ITEM 64
1496  002056  055232                    EM64               ;RECALIBRATE COMMAND CAUSED
1497                                                        ;IMPROPER REGISTER CHANGE.
1498                                                        ;GOOD DATA GIVES WHAT SHOULD BE
1499                                                        ;THERE.
1500                                                        ;RECEIVED DATA GIVES WHAT WAS THERE
1501                                                        ;AFTER COMMAND
1502  002060  060260                    DH5
1503  002062  062252                    DT5
1504  002064  062546                    DF5
1505
1506
1507                              ;ITEM65
1508
1509  002066  055451                    EM65               ;INTERRUPT FAILING
1510  002070  061670                    DH65               ;PC
1511                                                        ;TEST NO
1512                                                        ;CONTENTS OF RHCS1
1513                                                        ;CONTENTS OF RHAS
1514                                                        ;CONTENTS OF RHDS1
```

```
1515  002072  062446                        DT65                    ;$ERRPC,TSTNM,CS1,AS,DS1
1516  002074  062632                        DF65                    ;0,0,0,0,0
1517
1518
1519                              ;ITEM66
1520  002076  055473                        EM66                    ;HEADER AND DATA COMMAND
1521                                                                  ;FOR HEAD SELECTION TEST
1522                                                                  ;CAUSED AN ERROR
1523                                                                  ;RHDST GIVES WHAT TRACK
1524                                                                  ;WAS BEING WRITTEN ON CYLINDER 0
1525                                                                  ;SECTOR 0
1526  002100  062004                        DH66                    ;PC
1527                                                                  ;RHDST
1528                                                                  ;RHER1
1529                                                                  ;RHER2
1530                                                                  ;RHER3
1531                                                                  ;RHCS1
1532                                                                  ;RHCS2
1533  002102  062462                        DT66                    ;$ERRPC,DST,ER1,ER2,ER3,CS1,CS2
1534  002104  062637                        DF66                    ;0,0,0,0,0,0,0
1535                              ;ITEM67
1536  002106  055704                        EM67                    ;READ HEADER AND DATA ERROR
1537                                                                  ;IN HEAD SELECTION TEST
1538                                                                  ;FIRST FOUR WORDS GIVE HEADER
1539                                                                  ;NEXT WORDS ARE DATA
1540                                                                  ;GOOD DATA WORDS GIVE
1541                                                                  ;THE TRACK NUMBER IN
1542                                                                  ;BITS 4,5,6,7,8
1543  002110  061115                        DH30
1544  002112  062352                        DT30
1545  002114  062601                        DF30
1546                              ;ITEM70
1547  002116  056160                        EM70                    ;READ HEADER AND DATA ERROR
1548                                                                  ;IN DIFFERENCE LINE TEST
1549                                                                  ;WORD NOS. 1-4 GIVE
1550                                                                  ;HEADER
1551                                                                  ;WORD NOS. 5-260 GIVE DATA
1552                                                                  ;WHICH IS THE CYLINDER
1553                                                                  ;ADDRESS
1554  002120  061115                        DH30
1555  002122  062352                        DT30
1556  002124  062601                        DF30
1557
1558                              ;ITEM 71
1559  002126  056366                        EM71                    ;FORCING OPI CAUSED IMPROPER REGISTER
1560                                                                  ; CHANGE
1561                                                                  ;GOOD DATA GIVES WHAT SHOULD
1562                                                                  ;BE THERE
1563                                                                  ;RECEIVED DATA GIVES WHAT WAS
1564                                                                  ;THERE AFTER 3 INDEX PULSES
1565  002130  060260                        DH5                     ;PC
1566                                                                  ;REG. ADDR.
1567                                                                  ;GOOD DATA
1568                                                                  ;RECEIVED DATA
1569  002132  062252                        DT5                     ;$ERRPC,REGADR,$GDDAT,$BDDAT
1570  002134  062546                        DF5                     ;0,0,0,0
```

```
1571
1572                                    ;ITEM72
1573  002136  056627                            EM72                    ;THERE WAS AN ERROR
1574                                                                    ;AFTER A WRITE HEADER
1575                                                                    ;AND DATA COMMAND
1576
1577  002140  062102                            DH72                    ;PC
1578                                                                    ;RHCS1
1579                                                                    ;RHCS2
1580                                                                    ;RHDS1
1581                                                                    ;RHDST
1582                                                                    ;RHCA
1583                                                                    ;RHER1
1584                                                                    ;RHWC
1585  002142  062504                            DT72                    ;$ERRPC,CS1,CS2,DS1,DST,CA,ER1,WC
1586  002144  062650                            DF72                    ;0,0,0,0,0,0,0,0
1587
1588
1589
1590
1591
1592                                    ;ITEM73
1593  002146  057075                            EM73                    ;READING OVER 3 INDEX
1594                                                                    ;PULSES CAUSED SC
1595  002150  062102                            DH72
1596  002152  062504                            DT72
1597  002154  062650                            DF72
1598
1599                                    ;ITEM74
1600  002156  057245                            EM74                    ;READING OVER 3 INDEX
1601                                                                    ;PULSES CAUSED OPI
1602  002160  062102                            DH72
1603  002162  062504                            DT72
1604  002164  062650                            DF72
1605
```

```
                                ;;*****************************************************************
                                ;RH11 REGISTER BITS
                                ;;*****************************************************************


002166  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 REGISTER BITS
                        ;;*****************************************************************


                        ;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)
```

```
                        TRK10=  40000                   ;TRACK FIELD 4 (BIT #14)
                        TRK20=  100000                  ;TRACK FIELD 5 (BIT #15)

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

                        WCU=    1                       ;WRITE CURRENT UNSAFE (BIT #0)
                        CSF=    2                       ;CURRENT SINK FAILURE (BIT #1)
                        WSU=    4                       ;WRITE SELECT UNSAFE (BIT #2)
                        CSU=    10                      ;CURRENT SWITCH UNSAFE (BIT #3)
                        MSE=    20                      ;MOTOR SEQUENCE ERROR (BIT #4)
                        TDF=    40                      ;TRANSITIONS DETECTOR FAILURE (BIT #5)
                        TUF=    100                     ;TRANSITIONS UNSAFE (BIT #6)
                        FEN=    200                     ;FAILSAFE ENABLED (BIT #7)
                        WRU=    400                     ;WRITE READY UNSAFE (BIT #8)
                        MHS=    1000                    ;MULTIPLE HEAD SELECT (BIT #9)
                        NHS=    2000                    ;NO HEAD SELECTION (BIT #10)
                        IXE=    4000                    ;INDEX ERROR (BIT #11)
                        VU30=   10000                   ;30VOLT UNSAFE (BIT #12)
                        PLU=    20000                   ;PLO UNSAFE (BIT #13)
                        ACU=    100000                  ;ACUNSAFE (BIT #15)

                        ;OFFSET REGISTER (RHOF) (#11)

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

                        OFREV=  200                     ;OFFSET NEGATIVE (REVERSE) (BIT #5)
                        HCI=    2000                    ;HEADER COMPARE INHIBIT (BIT #10)
                        ECI=    4000                    ;ERROR CORRECTION CODE INHIBIT (BIT #11)
                        FMT22=  10000                   ;FORMAT BIT (BIT #12

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



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



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



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

                        PSU=    1                       ;PACK SPEED UNSAFE (BIT #0)
```

```
1821            000002                  VUF=    2                       ;VELOCITY UNSAFE (BIT #1)
1822            000010                  UWR=    10                      ;ANY UNSAFE EXCEPT READ/WRITE (BIT #3)
1823            000020                  PRE=    20                      ;DISK PACK ROTATION ERROR (BIT #4)
1824            000040                  ACL=    40                      ;AC LOW (BIT #5)
1825            000100                  DCL=    100                     ;DC LOW (BIT #6)
1826            040000                  SKI=    40000                   ;SEEK INCOMPLETE (BIT #14)
1827            100000                  OCYL=   100000                  ;OFF CYLINDER (BIT #15)
1828
1829
1830
1831                                    ;ECC POSITION REGISTER (RHEC1) (#16)
1832                                    ;EACH BIT IS CALLED BY BIT NUMBER
1833
1834
1835
1836
1837                                    ;ECC PATTERN REGISTER (RHEC2) (#17)
1838                                    ;EACH BIT IS CALLED BY BIT NUMBER
1839
```

```
                                .SBTTL  REGISTER ADDRESSES



                                ;RP04/5/6 DISK I/O REGISTERS LOCATED IN THE RH11 CONTROLLER

002170  176722                  RHDB:   176722                  ;DATA BUFFER
002172  176702                  RHWC:   176702                  ;WORD COUNT
002174  176704                  RHBA:   176704                  ;BUS ADDRESS
002176  176710                  RHCS2:  176710                  ;CONTROL AND STATUS 2

                                ;RP04/5/6 DISK I/O REGISTERS LOCATED IN THE DEVICE CONTROL LOGIC (DCL)

002200  176700                  RHCS1:  176700                  ;CONTROL AND STATUS 1
002202  176714                  RHER1:  176714                  ;ERROR #1
002204  176706                  RHDST:  176706                  ;DESIRED SECTOR/TRACK ADDRESS
002206  176740                  RHER2:  176740                  ;ERROR #2
002210  176732                  RHOF:   176732                  ;OFFSET
002212  176734                  RHCA:   176734                  ;DESIRED CYLINDER ADDRESS
002214  176742                  RHER3:  176742                  ;ERROR #3
002216  176716                  RHAS:   176716                  ;ATTENTION SUMMARY
002220  176724                  RHMR:   176724                  ;MAINTAINABILITY
002222  176712                  RHDS1:  176712                  ;DRIVE STATUS
002224  176726                  RHDT:   176726                  ;DRIVE TYPE
002226  176730                  RHSN:   176730                  ;SERIAL NUMBER
002230  176744                  RHEC1:  176744                  ;ECC POSITION
002232  176746                  RHEC2:  176746                  ;ECC PATTERN
002234  176736                  RHCC:   176736                  ;CURRENT CYLINDER ADDRESS
002236  176720                  RHLA:   176720                  ;LOOK-AHEAD

                                ;ADDITIONAL REGISTERS LOCATED IN THE RH70 CONTROLLER LOGIC

002240  176750                  RHBAE:  176750                  ;BUS ADDRESS EXTENSION REGISTER
002242  176752                  RHCS3:  176752                  ;CONTROL AND STATUS REGISTER #3


                                ;P-CLOCK (KW11-P) I/O REGISTERS

002244  172540                  PCLCSR: 172540                  ;CONTROL AND STATUS REGISTERS
002246  172542                  PCLBUF: 172542                  ;COUNT SET BUFFER
002250  172544                  PCLCTR: 172544                  ;COUNTER

```

```
1883
1884
1885                                    ;THE FOLLOWING LOCATIONS ARE RESERVED FOR REGISTER SAVES
1886                                    ;ANY TIME THERE IS AN ERROR ALL THESE WILL BE FILLED
1887                                    ;ONLY SOME MAY BE PRINTED BUT ALL WILL BE FILLED TRUE
1888                                    ;FOR THE TIME JUST AFTER THE "ERROR" ERROR COMMAND
1889
1890  002252  000000                    DB:     0                       ;DATA BUFFER
1891  002254  000000                    WC:     0                       ;WORD COUNT
1892  002256  000000                    BA:     0                       ;BUS ADDRESS
1893  002260  000000                    CS2:    0                       ;CONTROL AND STATUS 2
1894
1895
1896  002262  000000                    CS1:    0                       ;CONTROL AND STATUS 1
1897  002264  000000                    ER1:    0                       ;ERROR #1
1898  002266  000000                    DST:    0                       ;DESIRED SECTOR/TRACK ADDRESS
1899  002270  000000                    ER2:    0                       ;ERROR #2
1900  002272  000000                    OF:     0                       ;OFFSET
1901  002274  000000                    CA:     0                       ;DESIRED CYLINDER ADDRESS
1902  002276  000000                    ER3:    0                       ;ERROR #3
1903  002300  000000                    AS:     0                       ;ATTENTION SUMMARY
1904  002302  000000                    MR:     0                       ;MAINTAINABILITY
1905  002304  000000                    DS1:    0                       ;DRIVE STATUS
1906  002306  000000                    DT:     0                       ;DRIVE TYPE
1907  002310  000000                    SN:     0                       ;SERIAL NUMBER
1908  002312  000000                    EC1:    0                       ;ECC POSITION
1909  002314  000000                    EC2:    0                       ;ECC PATTERN
1910  002316  000000                    CC:     0                       ;CURRENT CYLINDER ADDRESS
1911  002320  000000                    LA:     0                       ;LOOK-AHEAD
1912
1913
1914
```

```
1915
1916
1917
1918                                    ;FUNCTION EQUATES
1919
1920                                        ;*TABLE OF FUNCTIONS FOR RHCS1 THEN "GO" BIT HAS TO BE SET
1921  002322                            FUTABL:
1922  002322  000000                    NOPERA: 0                       ;NO OPERATION
1923  002324  000002                    UNLOAD: 2                       ;UNLOAD (STAND BY)
1924  002326  000006                    RECALI: 6                       ;RECALIBRATE
1925  002330  000010                    DCLEAR: 10                      ;DRIVE CLEAR
1926  002332  000012                    RELEAS: 12                      ;RELEASE (DUAL-PORT OPERATION)
1927  002334  000030                    SERCH:  30                      ;SEARCH COMMAND
1928  002336  000050                    WRCHEK: 50                      ;WRITE CHECK DATA
1929  002340  000052                    WRCHDT: 52                      ;WRITE CHECK HEADER AND DATA
1930  002342  000060                    WRIDAT: 60                      ;WRITE DATA
1931  002344  000062                    WRIFOR: 62                      ;WRITE HEADER AND DATA (FORMAT)
1932  002346  000070                    READAT: 70                      ;READ DATA
1933  002350  000072                    REFOR:  72                      ;READ HEADER AND DATA
1934  002352  000004                    SEECOM: 4                       ;SEEK COMMAND
1935  002354  000014                    OFSETC: 14                      ;OFFSET COMMAND
1936  002356  000016                    RETCL:  16                      ;RETURNTO CENTERLINE
1937  002360  000022                    PKACK:  22                      ;PACK ACKNOWLEDGE
1938  002362  000020                    READIN: 20                      ;READ IN
1939 .002364  000000                    ILLEGL: .WORD   0               ;COMPUTED ILLEGAL FUNCTION
1940
1941
1942
1943                                        ;*DATA BUFFER FOR READ WRITE
1944
1945
1946  002370  000422                    WRFROM: .BLKW   274.            ;WRITE FROM THIS BUFFER
1947  003434  000422                    REINTO: .BLKW   274.            ;READ INTO THIS BUFFER
1948
```

```
                                        ;RESERVED LOCATIONS FOR FLAGS AND INTERNAL PROGRAM CONTROL WORDS

004500  000000                REGADR: 0                       ;SAVE REGISTER ADDRESS HERE
004502  000000                ERWORD: 0                       ;SAVE ERROR WORD NUMBER HERE
004504  000000                TSTNM:  0                       ;TEST NUMBER
004506  000000                RP4VEC: 0                       ;CONTAINS ADDRESS OF LOCATION
                                                            ;WHERE AN RP04 INTERRUPT IS TO VECTOR TO
                                                            ;THIS MUST BE MOVED INTO 'RPVEC' TO BE
                                                            ;EFFECTIVE.

004510  000000                OFSTVL: 0                       ;OFFSET VALUE USED IN OFFSET TEST


004512  000024                SAVERE: .BLKW   20.             ;BLOCK TO SAVE REGISTERS
004562  000000                FINALA: 0                       ;SAVE LOOK AHEAD REGISTER AT END OF OPERATION
004564  000000                FINACC: 0                       ;SAVE CURRENT CYLINDER REGISTER AT END OF OPERATION


                                        ;TABLE FOR ATTENTION BITS
                                        ;ATTENTION TABLE

004566    001   002   004     ATABLE: .BYTE   1,2,4,10,20,40,100,200
004571    010   020   040
004574    100   200


                                        ;RESERVED LOCATIONS FOR UNIT SELECT

004576  000010                UNITS:  .BLKW   8.              ;THIS IS FILLED WITH -1
004616  000000                UNIT:   .WORD   0               ;UNIT UNDER TEST
004620  000000                NOUNIT: .WORD   0               ;NUMBER OF UNITS PRESENT
                                                            ;USED TO KEEP TRACK OF UNIT UNDER TEST
004622  000000                NUNIT:  .WORD   0               ;USED TO DETERMIN IF THERE ARE MORE
                                                            ;THAN ONE UNIT
004624  000000                NOPUSH: 0                       ;ALL ONES INDICATE NONE OF THE OPERATOR
                                                            ;INTERVENTION TESTE WILL BE PERFORMED
004626  000000                SELECT: .WORD   0               ;ALL ONES INDICATE UNIT TO BE SELECTED
004630  000000                UNITSL: .WORD   0               ;UNIT NO. SELECTED


004632  000000                ERFLG$: 0                       ;ERROR FLAG

004634  000000                FIRST:  0                       ;IF ZERO WILL TYPE HEADER
                                                            ;IF ONES WILL NOT TYPE HEADER

004636  000000                RP06:   0                       ;DEVICE TYPE FLAG

004640  000000                RH70:   0                       ;IF 1, PROGRAM IS RUNNING ON AN RH70
                                                            ;IF 0, PROGRAM IS ON AN RH11

004642  000000                RUNCTR: .WORD   0               ;'RUN' LINE DELAY COUNTER TO BE USED
                                                            ;WHILE THE SILO IS FILLING
```

```
 2005
 2006  004644  000000                  ATTENT: 0                       ;ATTENTION BIT FOR PRESENT UNIT
 2007  004646  000000                  TOTALAT:        0               ;TOTAL ATTENTION BITS
 2008  004650  000000                  SILOSZ: .WORD   0               ;RH SILO SIZE
 2009
 2010  004652  000000                  TMP0:   .WORD   0               ;TEMP STORAGE
 2011  004654  000000                  TMP1:   .WORD   0
 2012  004656  000000                  TMP4:   .WORD   0               ;TEMP STORAGE
 2013  004660  000000                  TMP5:   .WORD   0               ;TEMP STORAGE
```

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


                                        .SBTTL  SETUP TESTS


004662  012737  177777  004624  BEGIN1: MOV     #-1,@#NOPUSH    ;JUMP OVER OPERATOR REQUIRED TESTS
004670  005037  004626                  CLR     @#SELECT        ;DO NOT SELECT UNIT
004674  000412                          BR      START
004676  012737  177777  004626  BEGIN2: MOV     #-1,@#SELECT    ;SELECT UNIT
004704  005037  004624                  CLR     @#NOPUSH        ;DO NOT JUMP OVER ANY TEST
004710  000404                          BR      START
004712  005037  004626          BEGIN:  CLR     @#SELECT        ;DO NOT SELECT UNIT
004716  005037  004624                  CLR     @#NOPUSH        ;DO NOT JUMP OVER ANY TEST
                                                                ;NORMAL RUN

004722                          START:
004722  000005                          RESET
                                .SBTTL  INITIALIZE THE COMMON TAGS
                                ;;CLEAR THE COMMON TAGS ($CMTAG) AREA
004724  012706  001100                  MOV     #$CMTAG,R6      ;;FIRST LOCATION TO BE CLEARED
004730  005026                          CLR     (R6)+           ;;CLEAR MEMORY LOCATION
004732  022706  001140                  CMP     #SWR,R6 ;;DONE?
004736  001374                          BNE     .-6             ;;LOOP BACK IF NO
004740  012706  001000                  MOV     #STACK,SP       ;;SETUP THE STACK POINTER
                                ;;INITIALIZE A FEW VECTORS
004744  012737  036624  000020          MOV     #$SCOPE,@#IOTVEC ;;IOT VECTOR FOR SCOPE ROUTINE
004752  012737  000340  000022          MOV     #340,@#IOTVEC+2 ;;LEVEL 7
004760  012737  041022  000030          MOV     #$ERROR,@#EMTVEC ;;EMT VECTOR FOR ERROR ROUTINE
004766  012737  000340  000032          MOV     #340,@#EMTVEC+2 ;;LEVEL 7
004774  012737  042546  000034          MOV     #$TRAP,@#TRAPVEC ;;TRAP VECTOR FOR TRAP CALLS
005002  012737  000340  000036          MOV     #340,@#TRAPVEC+2;LEVEL 7
005010  012737  042632  000024          MOV     #$PWRDN,@#PWRVEC ;;POWER FAILURE VECTOR
005016  012737  000340  000026          MOV     #340,@#PWRVEC+2 ;;LEVEL 7
005024  005037  001212                  CLR     $TIMES          ;;INITIALIZE NUMBER OF ITERATIONS
005030  005037  001214                  CLR     $ESCAPE         ;;CLEAR THE ESCAPE ON ERROR ADDRESS
005034  112737  000001  001115          MOVB    #1,$ERMAX       ;;ALLOW ONE ERROR PER TEST
005042  012737  005042  001106          MOV     #.,$LPADR       ;;INITIALIZE THE LOOP ADDRESS FOR SCOPE
005050  012737  005050  001110          MOV     #.,$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.
005056  013746  000004                  MOV     @#ERRVEC,-(SP)  ;;SAVE ERROR VECTOR
005062  012737  005116  000004          MOV     #64$,@#ERRVEC   ;;SET UP ERROR VECTOR
005070  012737  177570  001140          MOV     #DSWR,SWR       ;;SETUP FOR A HARDWARE SWICH REGISTER
005076  012737  177570  001142          MOV     #DDISP,DISPLAY  ;;AND A HARDWARE DISPLAY REGISTER
005104  022777  177777  174026          CMP     #-1,@SWR        ;;TRY TO REFERENCE HARDWARE SWR
005112  001012                          BNE     66$             ;;BRANCH IF NO TIMEOUT TRAP OCCURRED
                                                                ;;AND  THE HARDWARE SWR IS NOT = -1
005114  000403                          BR      65$             ;;BRANCH IF NO TIMEOUT
005116  012716  005124          64$:    MOV     #65$,(SP)       ;;SET UP FOR TRAP RETURN
005122  000002                          RTI
005124  012737  000176  001140  65$:    MOV     #SWREG,SWR      ;;POINT TO SOFTWARE SWR
005132  012737  000174  001142          MOV     #DISPREG,DISPLAY
005140  012637  000004          66$:    MOV     (SP)+,@#ERRVEC  ;;RESTORE ERROR VECTOR
```

```
2070
2071
2072
2073  005144  012737  000000  177776          MOV     #0,PS           ;SET PROCESSOR STATUS TO 0
2074  005152  012737  000200  000036          MOV     #200,@#TRAPVEC+2 ;TRAP PRIORITY = 4
2075  005160  013700  002166                  MOV     @#RPVEC,R0      ;GET RP VECTOR ADDRESS
2076  005164  012720  036534                  MOV     #RPVECT,(R0)+   ;THIS IS FOR UNTIMELY INTERRUPTS
2077  005170  012710  000340                  MOV     #340,(R0)       ;RP04 INTERRUPT SERVICE ROUTINE
2078                                                                   ;PRIORITY = 7
2079
2080  005174  004737  037562                  JSR     PC,@#$TKINT     ;INITIALIZE THE TTY KEYBOARD
2081  005200  005737  004634                  TST     @#FIRST         ;IS THIS FIRST TIME ROUND ?
2082  005204  001001                          BNE     1$              ;DO NOT GIVE HEADER IF NOT
2083  005206  000402                          BR      2$              ;GIVE HEADER IF SO
2084  005210  000137  006020          1$:     JMP     @#SND1          ;SKIP OVERALL PROGRAM HEADER
2085
2086  005214                          2$:
2087  005214  104401  005222                  TYPE    ,68$            ;;TYPE ASCIZ STRING
2088  005220  000435                          BR      67$             ;;GET OVER THE ASCIZ
2089                                  ;;68$:  .ASCIZ  <15><12>?RP04/5/6 FUNCTIONAL CONTROLLER TEST, PART II  - CZRJJ-D?
2090  005314                          67$:
2091  005314  104401  005322                  TYPE    ,70$            ;;TYPE ASCIZ STRING
2092  005320  000434                          BR      69$             ;;GET OVER THE ASCIZ
2093                                  ;;70$:  .ASCIZ  <15><12>/ALL DCL'S UNDER TEST MUST BE LOCKED ON CORRECT PORT/<15><12>
2094  005412                          69$:
2095
2096  005412  104401  005420                  TYPE    ,72$            ;;TYPE ASCIZ STRING
2097  005416  000424                          BR      71$             ;;GET OVER THE ASCIZ
2098                                  ;;72$:  .ASCIZ  <15><12>/PROGRAMMABLE DRIVES WILL NOT BE USED/
2099  005470                          71$:
2100  005470  104401  005476                  TYPE    ,74$            ;;TYPE ASCIZ STRING
2101  005474  000433                          BR      73$             ;;GET OVER THE ASCIZ
2102                                  ;;74$:  .ASCIZ  <15><12>/IF CHANGES ARE REQUIRED ON PORT SWITCH, A CYCLE UP/
2103  005564                          73$:
2104  005564  104401  005572                  TYPE    ,76$            ;;TYPE ASCIZ STRING
2105  005570  000436                          BR      75$             ;;GET OVER THE ASCIZ
2106                                  ;;76$:  .ASCIZ  <15><12>/SEQUENCE IS REQUIRED FOR STROBING THE PORT SELECT FLOP/<15><12>
2107  005666                          75$:
2108
2109  005666  104401  005674                  TYPE    ,78$            ;;TYPE ASCIZ STRING
2110  005672  000430                          BR      77$             ;;GET OVER THE ASCIZ
2111                                  ;;78$:  .ASCIZ  <15><12>/ALL DCL'S NOT UNDER TEST MUST BE SWITCHED OFF/
2112  005754                          77$:
2113  005754  104401  005762                  TYPE    ,80$            ;;TYPE ASCIZ STRING
2114  005760  000417                          BR      79$             ;;GET OVER THE ASCIZ
2115                                  ;;80$:  .ASCIZ  <15><12>/OR LOCKED ON THE OTHER PORT/
2116  006020                          79$:
2117
2118  006020  012737  177777  004634  SND1:   MOV     #-1,@#FIRST     ;NEXT TIME DO NOT GIVE HEADER
2119
2120                                  .SBTTL  GET VALUE FOR SOFTWARE SWITCH REGISTER
2121  006026  005737  000042                  TST     @#42            ;;ARE WE RUNNING UNDER XXDP/ACT?
2122  006032  001006                          BNE     64$             ;;BRANCH IF YES
2123  006034  023727  001140  000176          CMP     SWR,#SWREG      ;;SOFTWARE SWITCH REG SELECTED?
2124  006042  001005                          BNE     65$             ;;BRANCH IF NO
2125  006044  104406                          GTSWR                   ;;GET SOFT-SWR SETTINGS
```

```
 2126  006046  000403                          BR      65$
 2127  006050  112737  000001  001134  64$:    MOVB    #1,$AUTOB       ;;SET AUTO-MODE INDICATOR
 2128  006056                          65$:
 2129
 2130  006056  032777  010000  173054  RH70CK: BIT     #SW12,@SWR      ;LOOK TO SEE IF USING RH70
 2131  006064  001403                          BEQ     3$              ;IF SW12 = 0, SKIP NEXT
 2132  006066  012737  000001  004640          MOV     #1,@#RH70       ;IF SW12 = 1, CU IS AN RH70
 2133  006074                          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
                                        ;;*****************************************************************

006074  012737  033526  004506  2$:     MOV     #TIME2,@#RP4VEC ;MAKE RP04/5/6 INTERRUPTS GO TO 'TIME 2'
006102  012737  177777  041170          MOV     #-1,@#PRITEM    ;CLEAR PREVIOUS ITEM NUMBER

006110  005737  004626                  TST     @#SELECT        ;WAS IT A 200 START

006114  001442                          BEQ     TST1            ;BRANCH IF STARTING FROM 200

006116  104401  006124                  TYPE    ,65$            ;;TYPE ASCIZ STRING
006122  000424                          BR      64$             ;;GET OVER THE ASCIZ
                                ;;65$:  .ASCIZ  <15><12>/SELECT UNIT NUMBER TO BE TESTED ?  /<15><12>
006174                          64$:
006174  104412                          RDOCT
006176  042716  177770                  BIC     #177770,(SP)    ;ONLY KEEP LAST 3 BITS
006202  011637  004616                  MOV     (SP),@#UNIT     ;SAVE UNIT TO BE TESTED
006206  012637  004630                  MOV     (SP)+,@#UNITSL  ;SAVE UNIT TO BE TESTED

006212  001403                          BEQ     TST1            ;BRANCH IF STARTING FROM 200

006214  013737  004630  004616          MOV     @#UNITSL,@#UNIT ;SET UNIT NUMBER
```

```
2179
2180                                    ;;*****************************************************************
2181                                    ;*TEST 1         REFERENCE EACH REGISTER
2182
2183                                    ;*              REFERENCE EACH REGISTER BY A MOVE INSTRUCTION
2184
2185                                    ;;*****************************************************************
2186  006222  000004                    TST1:   SCOPE
2187  006224  012737  000001  001212            MOV     #1,$TIMES       ;;DO 1 ITERATION
2188
2189  006232  012737  000001  004504            MOV     #2-1,@#TSTNM    ;THIS SAVES TEST NUMBER
2190
2191  006240  012706  001000                    MOV     #STACK, SP      ;SET UP STACK POINTER
2192  006244  012737  041032  000030            MOV     #REGSA1,@#EMTVEC;ERROR VECTOR SO THAT
2193                                                                    ;NO REGISTERS ARE SAVED
2194  006252  012737  006300  000004            MOV     #2$,@#ERRVEC    ;SET UP FOR BUS TIMEOUT
2195
2196  006260  012700  000024                    MOV     #24,R0          ;THERE ARE 24 REG TO TEST
2197  006264  012701  002170                    MOV     #RHDB,R1        ;R1 NOW HAS ADDR OF ADDR OF FIRST REG.
2198  006270  013102                    1$:     MOV     @(R1)+, R2      ;READ HARDWARE REG.
2199  006272  005300                            DEC     R0              ;COUNT DOWN
2200  006274  001375                            BNE     1$              ;BRANCH IF 24 NOT DONE
2201  006276  000454                            BR      3$              ;BRANCH IF 24 DONE
2202  006300  012737  000006  000004    2$:     MOV     #ERRVEC+2,@#ERRVEC ;RESTORE TRAP CATCHER
2203  006306  022626                            CMP     (SP)+,(SP)+     ;CLEAN STACK
2204  006310  016137  177776  001200            MOV     -2(R1), $TMP1   ;STORE FAILING REG ADDR
2205  006316  104007                            ERROR   7               ;REGISTER NON EXISTANT
2206  006320  032777  020000  172612            BIT     #SW13,@SWR      ;INHIBIT ERROR PRINTOUT ?
2207  006326  001036                            BNE     4$              ;BRANCH IF YES
2208
2209  006330  104401  006336                    TYPE    ,65$            ;;TYPE ASCIZ STRING
2210  006334  000427                            BR      64$             ;;GET OVER THE ASCIZ
2211                                    ;;65$:  .ASCIZ  <15><12>/TO CHANGE BASE ADDRESS, RESTART AT ADDRESS /
2212  006414                            64$:
2213
2214  006414  012746  000204                    MOV     #ADDMOD,-(SP)   ;GET READY TO TYPE STARTING ADDRESS
2215                                                                    ;OF "CHANGE OF BASE ADDRESS" ROUTINE
2216  006420  104402                            TYPOC
2217  006422  000000                            HALT
2218  006424  000137  033016            4$:     JMP     @#$EOP          ;GO TO END OF PROGRAM ---------->
2219
2220  006430  012737  006510  000004    3$:     MOV     #TRP,@#4        ;INITIALIZE VECTOR
2221  006436  005037  004640                    CLR     RH70            ;INIT RH INDICATOR ++ C.W
2222  006442  005777  173572                    TST     @RHBAE          ;ADDRESS RPBAE(RH11/RH70?)
2223  006446  005237  004640                    INC     RH70            ;FOUND AN RH70-SET MASK
2224  006452  104401  006460                    TYPE    ,67$            ;;TYPE ASCIZ STRING
2225  006456  000413                            BR      66$             ;;GET OVER THE ASCIZ
2226                                    ;;67$:  .ASCIZ  <15><12>/RH70 CONTROLLER   /
2227  006506                            66$:
2228  006506  000417                            BR      RTN             ;GET OUT
2229  006510  022626                    TRP:    CMP     (SP)+,(SP)+     ;ADJUST THE STACK
2230  006512  104401  006520                    TYPE    ,65$            ;;TYPE ASCIZ STRING
2231  006516  000413                            BR      64$             ;;GET OVER THE ASCIZ
2232                                    ;;65$:  .ASCIZ  <15><12>/RH11 CONTROLLER   /
2233  006546                            64$:
2234  006546  012737  041022  000030    RTN:    MOV     #$ERROR,@#EMTVEC;RESTORE ERROR VECTOR
```

```
2235                                                    ;SO THAT REGISTERS ARE SAVED
2236  006554  012737  000006  000004          MOV     #ERRVEC+2,@#ERRVEC ;RESTORE TRAP CATCHER
2237
2238                                          ;FIND THE SILO SIZE
2239
2240  006562  004737  033314                  JSR     PC,     @#CLDISK        ;CONTROLLER CLEAR
2241  006566  005037  004650                  CLR     SILOSZ                  ;CLEAR SILO COUNTER
2242  006572  013777  004650  173370  13$:    MOV     SILOSZ, @RHDB           ;LOAD SILO
2243  006600  005237  004650                  INC     SILOSZ                  ;KEEP COUNT
2244  006604  032777  000100  173364          BIT     #IR,    @RHCS2          ;IS THE SILO FULL?
2245  006612  001367                          BNE     13$                     ;BRANCH IF NO
2246  006614  012737  000412  015570          MOV     #266.,  VAR1            ;VAR1 IN TEST 15
2247  006622  163737  004650  015570          SUB     SILOSZ, VAR1
2248  006630  005437  015570                  NEG     VAR1
2249  006634  013737  015570  023336          MOV     VAR1,   VAR3            ;VAR3 IN TEST 25
2250  006642  012737  002370  015600          MOV     #WRFROM,        VAR2    ;VAR2 IN TEST 15
2251  006650  063737  004650  015600          ADD     SILOSZ, VAR2
2252  006656  063737  004650  015600          ADD     SILOSZ, VAR2
2253  006664  013737  015600  023346          MOV     VAR2,   VAR4            ;VAR4 IN TEST 25
2254  006672  022737  000406  004650          CMP     #262.,SILOSZ            ;RH70C?
2255  006700  001003                          BNE     20$                     ;BRANCH IF NO
2256  006702  012737  000001  015616          MOV     #1,     VAR5-4          ;VAR5 IN TEST 15
2257  006710  004737  033314          20$:    JSR     PC,     @#CLDISK        ;CONTROLLER CLEAR
```

```
2258
2259                                    ;;*****************************************************************
2260                                    ;*TEST 2        PARTIAL TEST OF RHAS FOR UNIT NUMBERS PRESENT
2261
2262                                    ;*      CHECK THAT RHAS CAN BE CLEARED BY MOVING ALL ONES
2263
2264                                    ;;*****************************************************************
2265  006714  000004                   TST2:   SCOPE
2266  006716  012737  000001  001212           MOV     #1,$TIMES       ;;DO 1 ITERATION
2267  006724  012706  001000                   MOV     #STACK,SP       ;SET STACK POINTER
2268
2269  006730  013701  002216                   MOV     @#RHAS,R1       ;R1 HAS ADDRESS OF RHAS
2270  006734  012711  177777                   MOV     #-1,@R1         ;WRITE ALL ONES INTO RHAS
2271  006740  105711                           TSTB    @R1             ;TEST IT FOR ALL 0'S
2272
2273  006742  001407                           BEQ     TST3            ;BRANCH IF GOOD
2274
2275  006744  011137  001126                   MOV     @R1,@#$BDDAT    ;BAD DATA
2276  006750  005037  001124                   CLR     @#$GDDAT        ;GOOD DATA
2277  006754  010137  004500                   MOV     R1,@#REGADR     ;FAILING REG. RHAS
2278  006760  104005                           ERROR   5               ;RHAS DOES NOT CLEAR BY WRITING ALL
2279                                                                   ;ONES INTO IT
2280
```

```
                                ;;*****************************************************************
                                ;*TEST 3         TEST FOR DRIVES PRESENT USING RHAS AND RHCS2
                                ;
                                ;*       THE NUMBER OF RP04/RP06 DRIVES PRESENT IS FOUND
                                ;*       BY MOVING ALL ONES INTO RHER1 WITH UNIT NUMBER
                                ;*       IN RHCS2 INCREMENTED FROM ZERO TO SEVEN.
                                ;
                                ;*       THE SET BITS IN RHAS WILL GIVE DRIVES PRESENT.
                                ;
                                ;*       THE DRIVE TYPE IS CHECKED TO BE RP04 OR RP06 AND
                                ;*       UNITS PRESENT ARE STORED IN A TABLE CALLED 'UNITS'
                                ;
                                ;;*****************************************************************
006762  000004                TST3:   SCOPE
006764  012737  000001  001212        MOV     #1,$TIMES       ;;DO 1 ITERATION
                                ;
006772  012737  000003  004504        MOV     #4-1,@#TSTNM    ;THIS SAVES TEST NUMBER
                                ;
007000  000005                        RESET                   ;START WITH AN INIT
007002  004737  037562                JSR     PC,@#$TKINT     ;INITILIZE THE TTY KEYBOARD
                                ;
007006  032777  020000  172124        BIT     #SW13,@SWR      ;INHIBIT ERROR TYPEOUT?
007014  001026                        BNE     4$              ;BRANCH IF YES
007016  104401  007024                TYPE    ,65$            ;;TYPE ASCIZ STRING
007022  000423                        BR      64$             ;;GET OVER THE ASCIZ
                                ;;65$:  .ASCIZ  <15><12><15><12>/LOOKING AT RHAS - DRIVES PRESENT/
007072                          64$:
007072  013701  002216          4$:     MOV     @#RHAS,R1       ;R1 HAS ADDR. OF RHAS
007076  013702  002176                  MOV     @#RHCS2,R2      ;R2 HAS ADDR. OF RHCS2
007102  005012                        CLR     @R2             ;CLEAR RHCS2
007104  012700  000010                MOV     #8.,R0          ;COUNT
007110  013704  002202                MOV     @#RHER1,R4      ;R4 HAS ADDR. OF RHER1
007114  012714  177777          1$:     MOV     #-1,@R4         ;MOVE ERRORS INTO RHER1
007120  005212                        INC     @R2             ;INCREMENT UNIT NO.
007122  005300                        DEC     R0              ;COUNT
007124  001373                        BNE     1$              ;BRANCH IF 8 NOT DONE
007126  111137  004646                MOVB    @R1,@#TOTALAT   ;SAVE TOTAL ATTENTION
                                                                ;USED IN DRIVE CLEAR TEST
007132  105037  004647                CLRB    @#TOTALAT+1     ;CLEAR UPPER BYTE
007136  105711                        TSTB    @R1             ;TEST FOR ANY DRIVES PRESENT
007140  001402                        BEQ     2$              ;IF NONE THERE - SAY SO
007142  000137  007524                JMP     XE2             ;SOME THERE - LOAD TABLE
                                ;
007146  032777  020000  171764  2$:     BIT     #SW13,@SWR      ;INHIBIT ERROR TYPE OUT?
007154  001402                        BEQ     3$              ;BRANCH IF NO
007156  000137  010126                JMP     SELTST          ;CHECK FOR SELECTED UNIT START AND LOAD
                                                                ;"UNITS" TABLE WITH DESIRED DRIVE IF SO
                                ;
007162                          3$:
007162  104401  007170                TYPE    ,67$            ;;TYPE ASCIZ STRING
007166  000421                        BR      66$             ;;GET OVER THE ASCIZ
                                ;;67$:  .ASCIZ  <15><12><15><12>/NO DRIVES PRESENT - RHAS = 0/
007232                          66$:
007232  104401  007240                TYPE    ,69$            ;;TYPE ASCIZ STRING
```

```
                                        BR      68$             ;;GET OVER THE ASCIZ
                                ;;69$:  .ASCIZ  <15><12>/WRITING ONES INTO RHER1 FOR ALL UNIT NUMBERS/
                                68$:
                                        TYPE    ,71$            ;;TYPE ASCIZ STRING
                                        BR      70$             ;;GET OVER THE ASCIZ
                                ;;71$:  .ASCIZ  <15><12>/DOES NOT SET ANY BIT IN RHAS SO ABORT PROGRAM/
                                70$:
                                        TYPE    ,73$            ;;TYPE ASCIZ STRING
                                        BR      72$             ;;GET OVER THE ASCIZ
                                ;;73$:  .ASCIZ  <15><12>/TO LOOP ON THIS TEST WO PRINTOUT SET SWITCHES 13,8,1 & 0<15><12
                                72$:

                                        JMP     @#$EOP          ;GO OUT ------------------------->


                                        ;*SET UP THE UNITS TABLE

                                XE2:
                                2$:     MOV     #8.,R0          ;COUNTER
                                        MOV     #UNITS,R3       ;POINTER
                                3$:     MOV     #-1,(R3)+       ;PRESET BLOCK TO ALL ONES
                                        DEC     R0              ;COUNT
                                        BNE     3$              ;BRANCH IF 8 NOT DONE
                                        MOV     #UNITS,R3       ;POINTER
                                        CLR     R5
                                        CLR     @#NOUNIT        ;NO. OF UNITS PRESENT
                                        MOV     #8.,R0          ;COUNTER
                                        MOV     @R1,@#$TMP0     ;TEMPORARY STORAGE
                                4$:     ROR     @#$TMP0         ;SET CARRY IF ONE IN  0 BIT
                                        BCC     5$              ;CHECK NEXT UNIT IF ONE NOT IN BIT 0

                                        MOV     R5,@RHCS2       ;INSERT UNIT NUMBER INTO UA BITS
                                        CMP     #24020,@RHDT    ;IS THIS A DUAL PORT RP04 ?
                                        BEQ     7$              ;TYPE THE UNIT NO. IF YES
                                        CMP     #2002( ,RHDT    ;IS THIS A SINGLE PORT RP04 ?
                                        BEQ     7$              ;TYPE UNIT NO. IF YES

                                ;;**********************************************************************
                                        CMP     #24021,@RHDT    ;DUAL PORT RP05 ?
                                        BEQ     7$              ;TYPE UNIT NO. IF SO
                                        CMP     #20021,@RHDT    ;SINGLE PORT RP05 ?
                                        BEQ     7$              ;TYPE UNIT NO. IF SO

                                        CMP     #24022,@RHDT    ;IS THIS A DUAL PORT RP06 ?
                                        BEQ     7$              ;TYPE THE UNIT NO. IF SO
                                        CMP     #20022,@RHDT    ;IS THIS A SINGLE PORT RP06 ?
                                        BEQ     7$              ;TYPE UNIT NO. IF SO
                                        BR      9$
                                7$:     BIT     #BIT09, @RHDS1  ;IS THE DRIVE PROGRAMMABLE?
                                        BNE     8$              ;BRANCH IF YES
                                        BR      6$
                                8$:     TYPE    ,$CRLF
                                        MOV     R5,     -(SP)
                                        TYPDS                   ;TYPE THE DRIVE NUMBER
                                        TYPE    ,NOUSE          ;REPORT THIS DRIVE WILL NOT BE USED(DRIVE MUST BE LOCKED
                                        BR      5$
```

```
2393  007712                          9$:
2394                                  ;;*****************************************************************
2395
2396
2397  007712  104401  007720                TYPE    ,65$            ;;TYPE ASCIZ STRING
2398  007716  000410                        BR      64$             ;;GET OVER THE ASCIZ
2399                                ;;65$:  .ASCIZ  <15><12>/UNIT NUMBER /
2400  007740                        64$:
2401  007740  010546                        MOV     R5,-(SP)        ;GET READY TO TYPE UNIT NUMBER
2402  007742  104405                        TYPDS
2403  007744  104401  007752                TYPE    ,67$            ;;TYPE ASCIZ STRING
2404  007750  000406                        BR      66$             ;;GET OVER THE ASCIZ
2405                                ;;67$:  .ASCIZ  /,  RHDT = /
2406  007766                        66$:
2407  007766  017746  172232                MOV     @RHDT,-(SP)     ;GET READY TO TYPE RHDT
2408  007772  104402                        TYPOC
2409  007774  104401  010002                TYPE    ,69$            ;;TYPE ASCIZ STRING
2410  010000  000422                        BR      68$             ;;GET OVER THE ASCIZ
2411                                ;;69$:  .ASCIZ  ? - NOT AN RP04/RP05/RP06 DEVICE !!?
2412  010046                        68$:
2413  010046  000407                        BR      5$              ;NO RP04/RP05/RP06 FOUND SO INCR TABLE
2414
2415  010050  010523                6$:     MOV     R5,(R3)+
2416  010052  104401  001223                TYPE    ,$CRLF
2417  010056  010546                        MOV     R5,-(SP)
2418  010060  104405                        TYPDS                   ;TYPE DRIVE NO.
2419  010062  005237  004620                INC     @#NOUNIT        ;NUMBER OF DRIVES
2420
2421  010066  005205                5$:     INC     R5              ;INCR UNIT NUMBER
2422  010070  005300                        DEC     R0              ;DECR NO. OF UNITS LOOKED AT
2423  010072  001235                        BNE     4$              ;TEST THE NEXT UNIT
2424
2425  010074  005737  004620                TST     @#NOUNIT        ;IF THERE ARE ANY UNITS...
2426  010100  001002                        BNE     10$             ;CONTINUE
2427  010102  000137  033016                JMP     @#$EOP          ;ELSE GO TO END OF PASS
2428  010106                        10$:
2429
2430  010106  013737  004576  004616        MOV     @#UNITS,@#UNIT  ;SET UNIT NO. TO FIRST ONE FOUND/OR 0
2431  010114  013737  004620  004622        MOV     @#NOUNIT,@#NUNIT ;SAVE NO. OF UNITS
2432  010122  005337  004622                DEC     @#NUNIT         ;IF NUNIT = 0 THEN ONLY ONE UNIT
2433                                                                ;IF NUNIT > 0 THEN MORE THAN ONE UNIT
2434
2435  010126  005737  004626        SELTST: TST     @#SELECT        ;STARTING ADDRESS 200 ?
2436
2437  010132  001403                        BEQ     TST4            ;BRANCH IF STARTING FROM 200
2438
2439  010134  013737  004630  004616        MOV     @#UNITSL,@#UNIT ;SET UNIT NUMBER
```

```
2440
2441
2442                                    ;;*****************************************************************
2443                                    ;*TEST 4         TYPE SERIAL NUMBER AND DRIVE TYPE
2444
2445                                    ;*      SET APPROPRIATE ATTTENTION BIT OF UNIT UNDER TEST IN 'ATTENT'
2446                                    ;*      TYPE UNIT UNDER TEST
2447
2448                                    ;*      READ SERIAL NUMBER AND DRIVE TYPE REGISTERS
2449                                    ;*      TYPE THEM OUT AND PROCEED
2450
2451                                    ;*      TO LOOP HERE SET SWITCH 8, AND THIS TEST NUMBER ON
2452                                    ;*      SWITCHES 0 THRU 7 AND RESTART
2453
2454                                    ;;*****************************************************************
2455  010142  000004                    TST4:   SCOPE
2456  010144  012737  000001  001212            MOV     #1,$TIMES       ;;DO 1 ITERATION
2457  010152  012737  011044  001106            MOV     #1$,$LPADR      ;;SET SCOPE LOOP ADDRESS
2458
2459  010160  012706  001000            MOV     #STACK,SP       ;RESET STACK
2460  010164  012737  000004  004504            MOV     #4,@#TSTNM      ;SAVE TEST NUMBER
2461
2462  010172  004737  033314            JSR     PC,@#CLDISK     ;SET R1-RHCS1, R2-RHCS2
2463                                                            ;R3-RHDS1, R4-RHER1
2464                                                            ;GIVE RH-11 INITIALIZE
2465                                                            ;SETUP UNIT NUMBER
2466  010176  005037  004644            CLR     @#ATTENT        ;CLEAR UNIT UNDER TEST ATTENTION
2467
2468  010202  005737  004616            TST     @#UNIT          ;IS THE "UNIT" = 0 ?
2469  010206  001107                    BNE     20$             ;IF NOT, SKIP NEXT MODS
2470  010210  012700  000041            MOV     #41,R0          ;IF SO, CHECK THE LOAD MEDIA LOCATION
2471  010214  122710  000011            CMPB    #11,(R0)        ;WAS IT AN RP04/5/6 ?
2472  010220  001102                    BNE     20$             ;IF NOT, GO AHEAD AND TEST UNIT #0
2473  010222  005737  004626            TST     @#SELECT        ;WAS UNIT #0 SELECTED ?
2474                                                            ;(IE. 210 START ?)
2475  010226  001006                    BNE     19$             ;IF SO, CHANGE PACK
2476
2477                                    ;*INCREMENT THE UNITS TABLE TO NEXT DRIVE (IF ANY)
2478                                    ;*& DECREMENT "NOUNITS" PRESENT TO BE TESTED
2479
2480  010230  012700  004576            MOV     #UNITS,R0       ;IF NOT, LOAD THE UNITS TABLE POINTER
2481  010234  005720                    TST     (R0)+           ;SELECT THE NEXT UNIT IN TABLE
2482                                                            ;(DOUBLE INCREMENT THE POINTER, R0)
2483  010236  022710  177777            CMP     #-1,(R0)        ;IS THERE ANOTHER TABLE ENTRY PRESENT ?
2484  010242  001065                    BNE     18$             ;IF SO, USE NEXT DRIVE & DECR "NOUNITS"
2485                                                            ;IF NOT, CHANGE PACK ON UNIT #0
2486
2487  010244                    19$:
2488  010244  104401  010252            TYPE    ,65$            ;;TYPE ASCIZ STRING
2489  010250  000434                    BR      64$             ;;GET OVER THE ASCIZ
2490                            ;;65$:  .ASCIZ  <15><12><15><12>/DISMOUNT PACK FROM UNIT #0 AND MOUNT A SCRATCH PACK/
2491  010342                    64$:
2492  010342  104401  010350            TYPE    ,67$            ;;TYPE ASCIZ STRING
2493  010346  000421                    BR      66$             ;;GET OVER THE ASCIZ
2494                            ;;67$:  .ASCIZ  <15><12>/PRESS CONTINUE WHEN FINISHED/<15><12>
2495  010412                    66$:
```

```
2496  010412  000000                        HALT
2497  010414  000404                        BR      20$             ;CONTINUE, USING SCRATCH PACK ON UNIT #0
2498
2499  010416  011037  004616          18$:  MOV     (R0),@#UNIT     ;SET UP NEW UNIT UNDER TEST
2500  010422  005337  004620                DEC     @#NOUNITS       ;DECR BECAUSE UNIT #0 WON'T BE TESTED
2501
2502  010426  013700  004616          20$:  MOV     @#UNIT,R0       ;R0 NOW CONTAINS UNIT NO
2503
2504
2505
2506
2507                                  ;;**********************************************************************
2508  010432  005037  004636                CLR     @#RP06          ;CLEAR RP06 DEVICE TYPE FLAG
2509  010436  010077  171534                MOV     R0,@RHCS2       ;SET UP UNIT ADDRESSING
2510  010442  022777  024022  171554        CMP     #24022,@RHDT    ;DUAL PORT RP06 ?
2511  010450  001405                        BEQ     2$              ;YES..SET FLAG
2512  010452  022777  020022  171544        CMP     #20022,@RHDT    ;SINGLE PORT RP06 ?
2513  010460  001401                        BEQ     2$              ;YES...SET FLAG
2514  010462  000403                        BR      3$              ;DON'T SET RP06 FLAG
2515  010464  012737  177777  004636  2$:   MOV     #-1,@#RP06      ;SET THE FLAG
2516
2517  010472                          3$:                           ;ASSUME THE NEXT UNIT IS AN RP04/RP05
2518                                  ;;**********************************************************************
2519
2520
2521  010472  116037  004566  004644        MOVB    ATABLE(R0),@#ATTENT ;SET APPROPRIATE ATTENTION BIT
2522  010500  104401  010506                TYPE    ,69$            ;;TYPE ASCIZ STRING
2523  010504  000414                        BR      68$             ;;GET OVER THE ASCIZ
2524                                  ;;69$:  .ASCIZ  <15><12>/TESTING DRIVE NUMBER/
2525  010536                          68$:
2526  010536  013746  004616                MOV     @#UNIT,-(SP)    ;UNIT NO. TO STACK
2527  010542  104405                        TYPDS                   ;TYPE  DRIVE NO.
2528  010544  104401  010552                TYPE    ,71$            ;;TYPE ASCIZ STRING
2529  010550  000410                        BR      70$             ;;GET OVER THE ASCIZ
2530                                  ;;71$:  .ASCIZ  <15><12>/SERIAL NO. = /
2531  010572                          70$:
2532  010572  017746  171430                MOV     @RHSN,-(SP)     ;;SAVE @RHSN FOR TYPEOUT
2533  010576  104402                        TYPOC                   ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
2534  010600  104401  010606                TYPE    ,73$            ;;TYPE ASCIZ STRING
2535  010604  000410                        BR      72$             ;;GET OVER THE ASCIZ
2536                                  ;;73$:  .ASCIZ  <15><12>/DRIVE TYPE = /
2537  010626                          72$:
2538  010626  017746  171372                MOV     @RHDT,-(SP)     ;;SAVE @RHDT FOR TYPEOUT
2539  010632  104402                        TYPOC                   ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
```

```
2540
2541                                    ;*TYPE OUT THE DRIVE TYPE IN ASCII
2542
2543
2544                                    ;;*****************************************************************
2545  010634  022777  024020  171362          CMP     #24020,@RHDT    ;DUAL PORT RP04 ?
2546  010642  001424                        BEQ     4$              ;TYPE ASCII MESSAGE OUT
2547  010644  022777  020020  171352          CMP     #20020,@RHDT    ;SINGLE PORT RP04 ?
2548  010652  001420                        BEQ     4$              ;TYPE THE MESSAGE
2549
2550  010654  022777  024021  171342          CMP     #24021,@RHDT    ;DUAL PORT RP05 ?
2551  010662  001433                        BEQ     5$              ;TYPE THE MESSAGE
2552  010664  022777  020021  171332          CMP     #20021,@RHDT    ;SINGLE PORT RP05 ?
2553  010672  001427                        BEQ     5$              ;TYPE THE MESSAGE
2554
2555  010674  022777  024022  171322          CMP     #24022,@RHDT    ;DUAL PORT RP06 ?
2556  010702  001442                        BEQ     6$              ;TYPE THE MESSAGE
2557  010704  022777  020022  171312          CMP     #20022,@RHDT    ;SINGLE PORT RP06 ?
2558  010712  001436                        BEQ     6$              ;TYPE THE MESSAGE
2559
2560  010714                          4$:
2561  010714  104401  010722                  TYPE    ,75$            ;;TYPE ASCIZ STRING
2562  010720  000413                        BR      74$             ;;GET OVER THE ASCIZ
2563                                  ;;75$:  .ASCIZ  <15><12>/DRIVE IS AN RP04/<15><12>
2564  010750                          74$:
2565  010750  000435                        BR      1$              ;SKIP NEXT ONES
2566  010752                          5$:
2567  010752  104401  010760                  TYPE    ,77$            ;;TYPE ASCIZ STRING
2568  010756  000413                        BR      76$             ;;GET OVER THE ASCIZ
2569                                  ;;77$:  .ASCIZ  <15><12>/DRIVE IS AN RP05/<15><12>
2570  011006                          76$:
2571  011006  000416                        BR      1$              ;SKIP NEXT
2572  011010                          6$:
2573  011010  104401  011016                  TYPE    ,79$            ;;TYPE ASCIZ STRING
2574  011014  000413                        BR      78$             ;;GET OVER THE ASCIZ
2575                                  ;;79$:  .ASCIZ  <15><12>/DRIVE IS AN RP06/<15><12>
2576  011044                          78$:
2577                                  ;;*****************************************************************
2578
2579
2580
2581
2582  011044  005777  171156          1$:     TST     @RHSN           ;READ SERIAL NO. AND DRIVE TYPE
2583  011050  005777  171150                  TST     @RHDT           ;THESE TWO ARE TO HELP SCOPE LOOPS
2584  011054  017737  171146  002310          MOV     @RHSN,@#SN      ;SAVE TO CHECK IF DRIVE CLEAR CLEARS ANY BITS
2585  011062  017737  171136  002306          MOV     @RHDT,@#DT      ;SAVE TO CHECK IF DRIVE CLEAR CLEARS ANY BITS
```

```
2586
2587                                    ;;*****************************************************************
2588                                    ;*TEST 5        CHECK MOL TO BE HIGH
2589
2590                                    ;*      MAKE SURE THAT DRIVE IS ON LINE BEFORE STARTING PROGRAM
2591                                    ;*      IF DRIVE IS OFF LINE THEN AFTER TYPE OUT THE PROGRAM WILL
2592                                    ;*      HANG FOR EVER WAITING FOR DRIVE TO GO ON LINE
2593
2594                                    ;;*****************************************************************
2595  011070  000004                   TST5:   SCOPE
2596
2597  011072  012737  000005  004504            MOV     #6-1,@#TSTNM    ;THIS SAVES TEST NUMBER
2598
2599
2600
2601
2602  011100  004737  033314                    JSR     PC,@#CLDISK     ;GIVE INITILIZE
2603  011104  032713  010000                    BIT     #MOL,@R3        ;CHECK MOL IN RHDS1
2604
2605  011110  001144                            BNE     TST6            ;BRANCH IF MOL HIGH
2606
2607  011112  104401  011120                    TYPE    ,65$            ;;TYPE ASCIZ STRING
2608  011116  000420                            BR      64$             ;;GET OVER THE ASCIZ
2609                                    ;;65$:  .ASCIZ  <15><12>/DRIVE IS OFFLINE - MOL IS LOW/
2610  011160                            64$:
2611  011160  104401  011166                    TYPE    ,67$            ;;TYPE ASCIZ STRING
2612  011164  000424                            BR      66$             ;;GET OVER THE ASCIZ
2613                                    ;;67$:  .ASCIZ  <15><12>/HIT START ON DRIVE TO GET IT ON LINE/
2614  011236                            66$:
2615  011236  104401  011244                    TYPE    ,69$            ;;TYPE ASCIZ STRING
2616  011242  000431                            BR      68$             ;;GET OVER THE ASCIZ
2617                                    ;;69$:  .ASCIZ  <15><12>/PROGRAM WILL HANG TESTING MOL TILL MOL IS HIGH/
2618  011326                            68$:
2619  011326  032713  010000          1$:     BIT     #MOL,@R3        ;CHECK MOL IN RHDS1
2620  011332  001775                            BEQ     1$              ;WAIT IF MOL IS STILL LOW
2621  011334  104401  011342                    TYPE    ,71$            ;;TYPE ASCIZ STRING
2622  011340  000430                            BR      70$             ;;GET OVER THE ASCIZ
2623                                    ;;71$:  .ASCIZ  <15><12>/GOOD - MOL IS HIGH, PROGRAM WILL BE EXECUTED/
2624  011422                            70$:
2625
2626
2627
```

```
                                ;;*****************************************************************
                                ;*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
                                ;;*****************************************************************
011422  000004                  TST6:   SCOPE


011424  012737  000006  004504          MOV     #7-1,@#TSTNM    ;THIS SAVES TEST NUMBER

011432  012706  001000          .       MOV     #STACK,SP       ;RESET STACK

011436  004737  033314                  JSR     PC,@#CLDISK     ;SET R1-RHCS1, R2-RHCS2
                                                                ;R3-RHDS1, R4-RHER1
                                                                ;GIVE RH-11 INITIALIZE
                                                                ;SETUP UNIT NUMBER

011442  013700  002166                  MOV     @#RPVEC,R0      ;GET RP VECTOR ADDRESS
011446  012720  011514                  MOV     #RPTRP1,(R0)+   ;THIS IS FOR TIMELY INTERRUPTS
011452  012710  000340                  MOV     #340,(R0)       ;RP04 INTERRUPT SERVICE ROUTINE
                                                                ;PRIORITY = 7
011456  012737  000200  177776          MOV     #200,PS         ;SET PROCESSOR PRIORITY @ 4
011464  012711  000300                  MOV     #RDY!IE,@R1     ;RDY, IE IN RHSC1 SHOULD CAUSE INTERRUPT

011470  013737  034012  001200          MOV     @#TIMCNT,@#$TMP1;COUNTER
011476  005337  001200          1$:     DEC     @#$TMP1         ;WAIT FOR INTERRUPT
011502  001375                  .       BNE     1$              ;BRANCH IF NOT ZERO
                                                                ;BEFORE THIS IS ZERO INTERRUPT SHOULD OCCUR
011504  104065                          ERROR   65              ;INTERRUPT DID NOT OCCUR
011506  012712  000040                  MOV     #CLR,@R2        ;CLEAR CONTROLLER VIA CS2

011512  000407                  .       BR      TST7            ;BRANCH TO NEXT TEST --------------)


011514  022626          RPTRP1: CMP     (SP)+,(SP)+     ;RESTORE STACK
011516  022711  004200          .       CMP     #DVA!RDY,@R1    ;IE SHOULD BE LOW

011522  001403                  .       BEQ     TST7            ;CONTINUE IF GOOD --------------)

011524  104065                          ERROR   65              ;INTERRUPT OCCURED BUT
                                                                ;IE FAILED TO RESET
011526  012712  000040                  MOV     #CLR,@R2        ;CLEAR CONTROLLER
```

```
2674
2675
2676                            ;;*****************************************************************
2677                            ;*TEST 7        INTERRUPT AT PROCESSOR AND DISK PRIORITY SAME
2678
2679                            ;*      PROCESSOR PRIORITY IS SET AT 5 (SAME AS THE DISK)
2680                            ;*      IE AND RDY IS SET. THIS SHOULD NOT INTERRUPT
2681                            ;;*****************************************************************
2682  011532  000004            TST7:   SCOPE
2683
2684
2685  011534  012737  000007  004504        MOV     #10-1,@#TSTNM   ;THIS SAVES TEST NUMBER
2686
2687  011542  012706  001000                MOV     #STACK,SP       ;RESET STACK
2688
2689  011546  004737  033314                JSR     PC,@#CLDISK     ;SET R1-RHCS1, R2-RHCS2
2690                                                                ;R3-RHDS1, R4-RHER1
2691                                                                ;GIVE RH-11 INITIALIZE
2692                                                                ;SETUP UNIT NUMBER
2693
2694  011552  013700  002166                MOV     @#RPVEC,R0      ;GET RP VECTOR ADDRESS
2695  011556  012720  011622                MOV     #RPTRP2,(R0)+   ;THIS IS FOR UNTIMELY INTERRUPTS
2696  011562  012710  000340                MOV     #340,(R0)       ;RP04 INTERRUPT SERVICE ROUTINE
2697                                                                ;PRIORITY = 7
2698  011566  012737  000240  177776        MOV     #240,PS         ;SET PROCESSOR PRIORITY = 5
2699  011574  012711  000300                MOV     #RDY!IE,@R1     ;RDY, IE IN RHSC1 WHOULD CAUSE INTERRUPT
2700
2701  011600  013737  034012  001200        MOV     @#TIMCNT,@#$TMP1;COUNTER
2702  011606  005337  001200        1$:     DEC     @#$TMP1         ;WAIT FOR INTERRUPT
2703  011612  001375                BNE     1$              ;BRANCH IF NOT ZERO
2704                                                        ;BEFORE THIS IS ZERO INTERRUPT WHOULD OCCUR
2705  011614  012712  000040                MOV     #CLR,@R2        ;CLEAR CONTROLLER
2706
2707  011620  000404                BR      TST10           ;NO INTERRUPT SO CONTINUE ----------)
2708
2709
2710  011622  022626            RPTRP2: CMP     (SP)+,(SP)+     ;RESTORE STACK
2711  011624  104065                    ERROR   65              ;INTERRUPT OCCURRED WITH
2712                                                            ;PROCESSOR PRIORITY SAME AS DISK
2713  011626  012712  000040                MOV     #CLR,@R2        ;CLEAR CONTROLLER
2714
2715
2716
2717
```

```
2718
2719                                    ;;*****************************************************************
2720                                    ;*TEST 10        SET VV BIT #6 IN RHDS1
2721
2722                                    ;*      THIS TEST SETS VV IN RHDS1
2723                                    ;*      THERE IS A RESET AT THE BEGINING OF THE TEST
2724                                    ;*      FOR ERROR RECOVERY ONLY.
2725
2726                                    ;;*****************************************************************
2727  011632  000004                    TST10:  SCOPE
2728
2729                                            ;*IN CASE THERE IS ANY DRIVE ERRORS DURING POWER UP
2730                                            ;*OR POWER DOWN OR ANY PARITY ERRORS A RESET IS GIVEN
2731  011634  000005                    RESET
2732  011636  004737  037562            JSR     PC,@#$TKINT     ;INITILIZE TK
2733  011642  012737  000000  177776    MOV     #0,PS
2734
2735  011650  012706  001000            MOV     #STACK,SP       ;RESET STACK
2736  011654  012737  000010  004504    MOV     #10,@#TSTNM     ;SAVE TEST NUMBER
2737
2738  011662  004737  033314            JSR     PC,@#CLDISK     ;SET R1-RHCS1, R2-RHCS2
2739                                                            ;R3-RHDS1, R4-RHER1
2740                                                            ;GIVE RH-11 INITIALIZE
2741                                                            ;SETUP UNIT NUMBER
2742  011666  004737  033352            JSR     PC,@#CHECK      ;CHECK THAT DVA,RDY,MOL,DPR,DRY = 1
2743  011672  104401  057465            TYPE    ,CPHALT         ;CANNOT CONTINUE TESTS IF THEY DON'T
2744  011676  000000                    HALT                    ;STOP TESTING
2745
2746  011700  013777  002360  170272    MOV     @#PKACK,@RHCS1  ;GET READY FOR PKACK
2747                                                            ;PACK ACKNOWLEGDE WITH 22 IN RHCS1
2748
2749
2750                                            ;*NOW SAVE REGISTERS FOR COMPARISON AFTER PACK ACKNOWLEDGE
2751
2752  011706  004037  033462            JSR     R0,@#SAVER      ;SAVE REGISTERS
2753  011712  002172                    RHWC                    ;RHWC IS THE FIRST REGISTER SAVED
2754  011714  004512                    SAVERE                  ;STARTING ADDRES OF WHERE
2755                                                            ;THE REGISTERS ARE SAVED
2756  011716  000022                    18.                     ;NUMBER OF REGISTERS
2757                                                            ;SAVED = 18.
2758
2759  011720  013777  004506  170240    MOV     @#RP4VEC,@RPVEC ;SET RP04 VECTOR ADDRESS
2760                                                            ;TO 'TIME1' IF P-CLOCK IS PRESENT
2761                                                            ;OR TO 'TIME2' IF P-CLOCK IS NOT THERE
2762                                                            ;'TIME' WILL ONLY SAVE
2763                                                            ;CURRENT CYLINDER ADDRESS
2764                                                            ;AND LOOK AHEAD REGISTERS
2765
2766
2767  011726  013746  002360            MOV     @#PKACK,-(SP)   ;GET READY TO MOVE COMMAND
2768  011732  052716  000001            BIS     #GO,(SP)        ;GET READY TO SET GO
2769                                                            ;WITHOUT INTERRUPT ENABLE
2770  011736  012677  170236            MOV     (SP)+,@RHCS1    ;GO WITH
2771                                                             ;22 IN RHCS1 FOR PACK ACKNOWLEDGE
2772                                                            ;WITH INTERRUPT DISABLED
2773
```

```
 2774  011742  011100                          MOV     @R1,R0          ;SAVE RHCS1 DURING ABOVE OPERATION
 2775  011744  011305                          MOV     @R3,R5          ;SAVE RHDS1 DURING ABOVE OPERATION
 2776
 2777
 2778  011746  104413                          WAT                     ;WAIT FOR VV BIT TO SET
 2779  011750  002222                          RHDS1                   ;WAIT FOR RHDS1 REGISTER
 2780  011752  000100                          VV                      ;WAIT FOR VV BIT IN RHDS1 REGISTER
 2781  011754  000001                          1.                      ;ALLOW 10 MICRO SECONDS
 2782  011756  000001                          1.                      ;VV MUST SET BETWEEN
 2783                                                                   ;00 AND 20 MICRO SECONDS
 2784
 2785                                          ;*COMPARE CONTENTS OF RHCS1 AND RHDS1 ALREADY SAVED IN
 2786                                          ;*R0 AND R5 IMMEDIATELY AFTER GO IS ISSUED
 2787
 2788  011760  013746  002360                  MOV     @#PKACK,-(SP)   ;SAVE COMMAND
 2789  011764  052716  004200                  BIS     #DVA!RDY,(SP)   ;INCLUDE DVA!RDY
 2790  011770  011637  001124                  MOV     (SP),@#$GDDAT   ;SAVE FOR PRINTOUT
 2791  011774  022600                          CMP     (SP)+,R0        ;DURING ABOVE OPERATION ONLY DVA!RDY
 2792                                                                   ;AND COMMAND SHOULD BE SET
 2793  011776  001405                          BEQ     64$             ;BRANCH IF GOOD
 2794  012000  010037  001126                  MOV     R0,@#$BDDAT     ;BAD DATA
 2795  012004  010137  004500                  MOV     R1,@#REGADR     ;FAILING REGISTER RHCS1
 2796  012010  104021                          ERROR   21              ;DURING ABOVE OPERATION ONLY
 2797                                                                   ;COMMAND AND DVA!RDY SHOULD BE SET
 2798  012012  012746  010700          64$:    MOV     #MOL!DPR!DRY!VV,-(SP)   ;SAVE BITS SET DURING OPERATION IN RHDS1
 2799  012016  011637  001124                  MOV     (SP),@#$GDDAT   ;SAVE FOR PRINTOUT
 2800  012022  022605                          CMP     (SP)+,R5        ;DURING ABOVE OPERATION ONLY MOL!DPR!DRY!VV
 2801                                                                   ;SHOULD BE SET
 2802  012024  001405                          BEQ     66$             ;BRANCH IF GOOD
 2803  012026  010537  001126                  MOV     R5,@#$BDDAT     ;BAD DATA
 2804  012032  010337  004500                  MOV     R3,@#REGADR     ;FAILING REGISTER RHDS1
 2805  012036  104063                          ERROR   63              ;DURING ABOVE OPERATION ONLY
 2806                                                                   ;MOL!DPR!DRY!VV SHOULD BE SET
 2807  012040                          66$:
 2808
 2809  012040  004037  034204                  JSR     R0,@#CHREG      ;CHANGE BITS IN SAVED REGISTER
 2810  012044  002222                          RHDS1                   ;CHANGE RHDS1 REGISTER
 2811
 2812  012046  000001                          1                       ;1 BIT/BITS TO BE CHANGED
 2813  012050  000001                          1                       ;NEW VALUE OF VV IS 1
 2814  012052  000100                          VV                      ;CHANGE VV BIT
 2815
 2816                                          ;*NOW COMPARE REGISTERS BEFORE PACK ACKNOWLEDGE
 2817                                          ;*WITH AFTER PACK ACKNOWLEDGE
 2818
 2819
 2820
 2821  012054  004037  034312                  JSR     R0,@#COMREG     ;COMPARE SAVED REGISTERS WITH
 2822                                                                   ;PRESENT VALUE
 2823  012060  004512                          SAVERE                  ;GOOD DATA SAVED IN 'SAVERE'
 2824  012062  002254                          WC                      ;TEST DATA STARTING FROM 'RHWC'
 2825  012064  000022                          18.                     ;18. REGISTERS TO BE COMPARED
 2826  012066  012072                          1$                      ;RETURN TO 1$ ON ERROR
 2827  012070  012076                          2$                      ;RETURN TO 2$ ON NO ERROR
 2828
 2829
```

```
 2830  012072  104015                  1$:     ERROR   15              ;GIVING A PACK ACKNOWLEDGE
 2831  012074  000207                          RTS     PC              ;CAUSED AN ERROR
 2832                                                                  ;PACK ACKNOWLEDGE SHOULD
 2833                                                                  ;SET VV IN RHDS1
 2834                                                                  ;INTERRUPT SHOULD MAKE
 2835                                                                  ;IE = 0
 2836                                                                  ;NO OTHER REGISTERS SHOULD
 2837                                                                  ;CHANGE
 2838                                                                  ;GOOD DATA GIVES CONTENTS
 2839                                                                  ;OF REGISTER BEFORE COMMAND
 2840                                                                  ;RECEIVED DATA GIVES CONTENTS
 2841                                                                  ;OF REGISTER AFTER COMMAND
 2842  012076                          2$:
 2843
 2844
 2845
 2846
```

```
                                .SBTTL  DATA TRANSFER RELATED ERRORS (USING MEDIA)


                                ;;*****************************************************************
                                ;*TEST 11       LAST BLOCK TRANSFERED-RHDS1 LBT

                                ;*      WRITE ONE WORD OF 65125 ON CYLINDER 410./814., TRACK 18
                                ;*      SECTOR 21, BY A WRITE HEADER AND DATA COMMAND
                                ;*      THEN CHECK ALL REGISTERS (LAST BLOCK TRANSFERED
                                ;*      SHOULD BE SET)

                                ;*      THEN READ ABOVE USING READ DATA (256 WORDS)
                                ;*      CHECK ALL REGISTERS AND DATA
                                ;*      (AGAIN 'LBT' SHOULD BE SET)

                                ;;*****************************************************************
012076  000004                  TST11:  SCOPE
012100  012706  001000                  MOV     #STACK,SP       ;RESET STACK
012104  012737  000011  004504          MOV     #11,@#TSTNM     ;SAVE TEST NUMBER

012112  004737  033314                  JSR     PC,@#CLDISK     ;SET R1-RHCS1, R2-RHCS2
                                                                ;R3-RHDS1, R4-RHER1
                                                                ;GIVE RH-11 INITIALIZE
                                                                ;SETUP UNIT NUMBER

                                        ;*CHECK THE DRIVE TYPE AND THEN FILL THE
                                        ;*WRITE FROM BUFFER WITH APPROPRIATE HEADER
                                ;;*****************************************************************
012116  005737  004636                  TST     @#RP06  ;TEST FOR RP06 DRIVE
012122  001412                          BEQ     11$             ;TREAT UNIT AS AN RP04
                                                                ;TREAT UNIT AS AN RP06

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


012124  004037  033140          JSR     R0,@#FLHEAD     ;SAVE HEADER DATA IN WRFROM
012130  002370                  WRFROM                  ;LOCATION WHERE SAVED
012132  000005                  5                       ;NUMBER OF WORDS SAVED
012134  011456                  11456                   ;FIRST DATA WORD
012136  011025                  <18.*400>!<21.>                 ;SECOND DATA WORD
012140  000000                  0                       ;THIRD DATA WORD
012142  000000                  0                       ;FOURTH DATA WORD
012144  065125                  <26.*2000>!<18.*40>!<21.>               ;FIFTH DATA WORD
012146  000411                  BR      12$             ;CONTINUE WITH SET UP

012150                  11$:

012150  004037  033140          JSR     R0,@#FLHEAD     ;SAVE HEADER DATA IN WRFROM
012154  002370                  WRFROM                  ;LOCATION WHERE SAVED
012156  000005                  5                       ;NUMBER OF WORDS SAVED
012160  010632                  10632                   ;FIRST DATA WORD
012162  011025                  <18.*400>!<21.>                 ;SECOND DATA WORD
012164  000000                  0                       ;THIRD DATA WORD
012166  000000                  0                       ;FOURTH DATA WORD
012170  065125                  <26.*2000>!<18.*40>!<21.>               ;FIFTH DATA WORD
012172                  12$:
```

```
 2903                                          ;*FILL READ INTO BUFFER WITH ALL ONES
 2904
 2905
 2906  012172  004037  033164                 JSR     R0,@#CLAREA     ;CLEAR 256 WORDS, FROM REINTO
 2907  012176  003434                         REINTO                  ;STARTING FROM REINTO
 2908  012200  000256                         256                     ;256 WORDS
 2909  012202  177777                         -1                      ;FILL WITH -1
 2910
 2911
```

```
                                        ;*DRIVE TYPE IS CHECKED AND THEN THE APPROPRIATE
                                        ;*WRITE HEADER AND DATA COMMAND IS LOADED
                                        ;
                                    ;;*********************************************************************
012204  005737  004636                TST     @#RP06  ;TEST FOR RP06 DRIVE
012210  001412                      BEQ     7$              ;TREAT UNIT AS RP04
                                    ;;*********************************************************************


012212  004037  035276                JSR     R0,@#RUN        ;SETUP TO RUN FOR DATA COMMAND
012216  001456                      814.                    ;CYLINDER 814.
012220     025                      .BYTE   21.             ;SECTOR 21.
012221     022                      .BYTE   18.             ;TRACK 18.
012222  177773                      -1-4                    ;WORD COUNT (DATA) = 1 +
                                                            ;4 HEADER WORDS
012224  002370                      WRFROM                  ;BUS ADDRESS
                                                            ;STARTING ADDRESS OF DATA
                                                            ;BUFFER = WRFROM
012226  000000                      0                       ;DO NOT INHIBIT BUS ADDRESS INCREMENT
012230  010000                      FMT22                   ;16 BITS PER WORD FORMAT
                                                            ;DO NOT INHIBIT ECC CORRECTION
                                                            ;DO NOT INHIBIT HEADER COMPARE
012232  002344                      WRIFOR                  ;GET READY TO DO A WRIFOR
                                                        ;WRITE HEADER AND DATA WITH 62 IN RHCS1

012234  000411                      BR      8$              ;CONTINUE WITH TEST

012236                          7$:

012236  004037  035276                JSR     R0,@#RUN        ;SETUP TO RUN FOR DATA COMMAND
012242  000632                      410.                    ;CYLINDER 410.
012244     025                      .BYTE   21.             ;SECTOR 21.
012245     022                      .BYTE   18.             ;TRACK 18.
012246  177773                      -1-4                    ;WORD COUNT (DATA) = 1 +
                                                            ;4 HEADER WORDS
012250  002370                      WRFROM                  ;BUS ADDRESS
                                                            ;STARTING ADDRESS OF DATA
                                                            ;BUFFER = WRFROM
012252  000000                      0                       ;DO NOT INHIBIT BUS ADDRESS INCREMENT
012254  010000                      FMT22                   ;16 BITS PER WORD FORMAT
                                                            ;DO NOT INHIBIT ECC CORRECTION
                                                            ;DO NOT INHIBIT HEADER COMPARE
012256  002344                      WRIFOR                  ;GET READY TO DO A WRIFOR
                                                        ;WRITE HEADER AND DATA WITH 62 IN RHCS1

012260                          8$:


                                        ;*NOW SAVE REGISTERS FOR COMPARISON AFTER WRITE

012260  004037  033462                JSR     R0,@#SAVER      ;SAVE REGISTERS
012264  002172                      RHWC                    ;RHWC IS THE FIRST REGISTER SAVED
012266  004512                      SAVERE                  ;STARTING ADDRES OF WHERE
                                                            ;THE REGISTERS ARE SAVED
012270  000022                      18.                     ;NUMBER OF REGISTERS
```

```
2968                                                    ;SAVED = 18.
2969
2970  012272  004737  033374                  JSR     PC,@#CHECKT     ;CHECKS DVA,RDY,MOL,DPR,DRY AND VV = 1
2971                                                    ;AND THAT ALL STATUS BITS ARE = 0
2972  012276  104401  057465                  TYPE    ,CPHALT         ;CANNOT CONTINUE TESTING IF NOT
2973  012302  000000                          HALT                    ;STOP TEST
2974  012304  013777  004506  167654          MOV     @#RP4VEC,@RPVEC ;SET RP04 VECTOR ADDRESS
2975                                                    ;TO 'TIME1' IF P-CLOCK IS PRESENT
2976                                                    ;OR TO 'TIME2' IF P-CLOCK IS NOT THERE
2977                                                    ;'TIME' WILL ONLY SAVE
2978                                                    ;CURRENT CYLINDER ADDRESS
2979                                                    ;AND LOOK AHEAD REGISTERS
2980
2981  012312  013746  002344                  MOV     @#WRIFOR,-(SP)  ;GET READY TO MOVE COMMAND
2982  012316  052716  000101                  BIS     #GO!IE,(SP)     ;GET READY TO SET GO AND
2983                                                    ;ENABLE INTERRUPT
2984  012322  012677  167652                  MOV     (SP)+,@RHCS1    ;GO WITH
2985                                                     ;62 IN RHCS1 FOR WRITE HEADER AND DATA
2986                                                    ;WITH INTERRUPT ENABLED
2987
2988                                          ;*TIME IS NOT CRITICAL HERE
2989
2990  012326  104413                          WAT                     ;WAIT FOR LBT BIT TO SET
2991  012330  002222                          RHDS1                   ;WAIT FOR RHDS1 REGISTER
2992  012332  002000                          LBT                     ;WAIT FOR LBT BIT IN RHDS1 REGISTER
2993  012334  004704                          2500.                   ;ALLOW 25000 MICRO SECONDS
2994  012336  004704                          2500.                   ;LBT MUST SET BETWEEN
2995                                                    ;00 AND 50000 MICRO SECONDS
2996
2997                                          ;*NOW CHANGE SAVED REGISTERS TO EXPECTED VALUE
2998
2999  012340  004037  033216                  JSR     R0,@#FILLRE     ;MOV 0 INTO SAVED RHWC
3000  012344  002172                          RHWC                    ;SAVED REGISTER TO CHANGE
3001  012346  000000                          0                       ;DATA
3002
3003
3004  012350  004037  033216                  JSR     R0,@#FILLRE     ;MOV WRFROM+<5*2> INTO SAVED RHBA
3005  012354  002174                          RHBA                    ;SAVED REGISTER TO CHANGE
3006  012356  002402                          WRFROM+<5*2>                    ;DATA
3007
3008
3009  012360  004037  034204                  JSR     R0,@#CHREG      ;CHANGE BITS IN SAVED REGISTER
3010  012364  002222                          RHDS1                   ;CHANGE RHDS1 REGISTER
3011
3012  012366  000001                          1                       ;1 BIT/BITS TO BE CHANGED
3013  012370  000001                          1                       ;NEW VALUE OF LBT IS 1
3014  012372  002000                          LBT                     ;CHANGE LBT BIT
3015
3016  012374  004037  034204                  JSR     R0,@#CHREG      ;CHANGE BITS IN SAVED REGISTER
3017  012400  002212                          RHCA                    ;CHANGE RHCA REGISTER
3018
3019  012402  000001                          1                       ;1 BIT/BITS TO BE CHANGED
3020  012404  000001                          1                       ;NEW VALUE OF BIT0 IS 1
3021  012406  000001                          BIT0                    ;CHANGE BIT0 BIT
3022
3023  012410  004037  033216                  JSR     R0,@#FILLRE     ;MOV 0 INTO SAVED RHDST
```

```
3024  012414  002204                          RHDST                   ;SAVED REGISTER TO CHANGE
3025  012416  000000                          0                       ;DATA
3026
3027
3028                                          ;*COMPARE ALL REGISTERS
3029
3030  012420  004037  034312                  JSR     R0,@#COMREG     ;COMPARE SAVED REGISTERS WITH
3031                                                                  ;PRESENT VALUE
3032  012424  004512                          SAVERE                  ;GOOD DATA SAVED IN 'SAVERE'
3033  012426  002254                          WC                      ;TEST DATA STARTING FROM 'RHWC'
3034  012430  000021                          17.                     ;17. REGISTERS TO BE COMPARED
3035  012432  012436                          1$                      ;RETURN TO 1$ ON ERROR
3036  012434  012442                          2$                      ;RETURN TO 2$ ON NO ERROR
3037
3038
3039  012436  104045                  1$:     ERROR   45              ;WRITING ON THE LAST BLOCK
3040  012440  000207                          RTS     PC              ;IE. CYLINDER 410./814., SECTOR 21
3041                                                                  ;TRACK 18 CAUSED
3042                                                                  ;IMPROPER REGISTER CHANGE
3043                                                                  ;GOOD DATA GIVES WHAT
3044                                                                  ;SHOULD BE THERE
3045                                                                  ;RECEIVED DATA GIVES WHAT
3046                                                                  ;WAS THERE AFTER WRITE
3047                                                                  ;ON LAST BLOCK
```

```
                                        ;*NOW A READ DATA WILL BE DONE ON SAME CYLINDER, SECTOR & TRACK
                                        ;*CLEAR ANY PREVIOUS ERRORS
012442                          2$:
012442  004737  033314                JSR     PC,@#CLDISK     ;SET R1-RHCS1, R2-RHCS2
                                                                ;R3-RHDS1, R4-RHER1
                                                                ;GIVE RH-11 INITIALIZE
                                                                ;SETUP UNIT NUMBER
                                        ;*FILL WRITE FROM BUFFER WITH EXPECTED DATA
012446  004037  033140                JSR     R0,@#FLHEAD     ;SAVE HEADER DATA IN WRFROM
012452  002370                        WRFROM                  ;LOCATION WHERE SAVED
012454  000001                        1                       ;NUMBER OF WORDS SAVED
012456  065125                        <26.*2000>!<18.*40>!<21.>               ;FIRST DATA WORD
012460  004037  033164                JSR     R0,@#CLAREA     ;CLEAR 256. WORDS, FROM WRFROM+2
012464  002372                        WRFROM+2                        ;STARTING FROM WRFROM+2
012466  000400                        256.                    ;256. WORDS
012470  000000                        0                       ;FILL WITH 0
                                        ;*FIRST THE DRIVE TYPE IS CHECKED AND THEN THE APPROPRIATE
                                        ;*READ COMMAND IS LOADED
                                ;;**************************************************************
012472  005737  004636                TST     @#RP06  ;TEST FOR RP06 DRIVE
012476  001412                        BEQ     9$              ;TREAT UNIT AS RP04
                                ;;**************************************************************
012500  004037  035276                JSR     R0,@#RUN        ;SETUP TO RUN FOR DATA COMMAND
012504  001456                        814.                    ;CYLINDER 814.
012506     025                      .BYTE   21.             ;SECTOR 21.
012507     022                      .BYTE   18.             ;TRACK 18.
012510  177400                      -256.                   ;WORD COUNT = 256.
012512  003434                      REINTO                  ;BUS ADDRESS
                                                                ;STARTING ADDRESS OF DATA
                                                                ;BUFFER = REINTO
012514  000000                      0                       ;DO NOT INHIBIT BUS ADDRESS INCREMENT
012516  010000                      FMT22                   ;16 BITS PER WORD FORMAT
                                                                ;DO NOT INHIBIT ECC CORRECTION
                                                                ;DO NOT INHIBIT HEADER COMPARE
012520  002346                      READAT                  ;GET READY TO DO A READAT
                                                        ;READ DATA WITH 70 IN RHCS1
012522  000411                      BR      10$             ;CONTINUE WITH TEST
012524                          9$:
012524  004037  035276                JSR     R0,@#RUN        ;SETUP TO RUN FOR DATA COMMAND
012530  000632                        410.                    ;CYLINDER 410.
012532     025                      .BYTE   21.             ;SECTOR 21.
```

```
3104  012533    022                          .BYTE   18.              ;TRACK 18.
3105  012534  177400                         -256.                   ;WORD COUNT = 256.
3106  012536  003434                         REINTO                  ;BUS ADDRESS
3107                                                                  ;STARTING ADDRESS OF DATA
3108                                                                  ;BUFFER = REINTO
3109  012540  000000                         0                       ;DO NOT INHIBIT BUS ADDRESS INCREMENT
3110  012542  010000                         FMT22                   ;16 BITS PER WORD FORMAT
3111                                                                  ;DO NOT INHIBIT ECC CORRECTION
3112                                                                  ;DO NOT INHIBIT HEADER COMPARE
3113  012544  002346                         READAT                  ;GET READY TO DO A READAT
3114                                                          ;READ DATA WITH 70 IN RHCS1
3115
3116  012546                          10$:
3117
3118                                        ;*SAVE REGISTERS FOR COMPARISON AFTER READ DATA
3119  012546  004037  033462                JSR     R0,@#SAVER      ;SAVE REGISTERS
3120  012552  002172                        RHWC                    ;RHWC IS THE FIRST REGISTER SAVED
3121  012554  004512                        SAVERE                  ;STARTING ADDRES OF WHERE
3122                                                                  ;THE REGISTERS ARE SAVED
3123  012556  000022                        18.                     ;NUMBER OF REGISTERS
3124                                                                  ;SAVED = 18.
3125  012560  004737  033374                JSR     PC,@#CHECKT     ;CHECKS DVA,RDY,MOL,DPR,DRY AND VV = 1
3126                                                                  ;AND THAT ALL STATUS BITS ARE = 0
3127  012564  104401  057465                TYPE    ,CPHALT         ;CANNOT CONTINUE TESTING IF NOT
3128  012570  000000                        HALT                    ;STOP TEST
3129  012572  013777  004506  167366        MOV     @#RP4VEC,@RPVEC ;SET RP04 VECTOR ADDRESS
3130                                                                  ;TO 'TIME1' IF P-CLOCK IS PRESENT
3131                                                                  ;OR TO 'TIME2' IF P-CLOCK IS NOT THERE
3132                                                                  ;'TIME' WILL ONLY SAVE
3133                                                                  ;CURRENT CYLINDER ADDRESS
3134                                                                  ;AND LOOK AHEAD REGISTERS
3135
3136  012600  013746  002346                MOV     @#READAT,-(SP)  ;GET READY TO MOVE COMMAND
3137  012604  052716  000101                BIS     #GO!IE,(SP)     ;GET READY TO SET GO AND
3138                                                                  ;ENABLE INTERRUPT
3139  012610  012677  167364                MOV     (SP)+,@RHCS1    ;GO WITH
3140                                                                   ;70 IN RHCS1 FOR READ DATA
3141                                                                  ;WITH INTERRUPT ENABLED
3142
3143                                        ;*TIME IS NOT CRITICAL HERE
3144
3145  012614  104413                        WAT                     ;WAIT FOR RDY BIT TO SET
3146  012616  002200                        RHCS1                   ;WAIT FOR RHCS1 REGISTER
3147  012620  000200                        RDY                     ;WAIT FOR RDY BIT IN RHCS1 REGISTER
3148  012622  001614                        908.                    ;ALLOW 9080 MICRO SECONDS
3149  012624  001502                        834.                    ;RDY MUST SET BETWEEN
3150                                                                  ;740 AND 17420 MICRO SECONDS
3151
3152                                        ;*NOW CHANGE SAVED REGISTERS TO EXPECTED VALUE
3153
3154
3155  012626  004037  033216                JSR     R0,@#FILLRE     ;MOV 0 INTO SAVED RHWC
3156  012632  002172                        RHWC                    ;SAVED REGISTER TO CHANGE
3157  012634  000000                        0                       ;DATA
3158
3159
```

```
3160  012636  004037  033216                JSR     R0,@#FILLRE     ;MOV REINTO+<256.*2> INTO SAVED RHBA
3161  012642  002174                       RHBA                    ;SAVED REGISTER TO CHANGE
3162  012644  004434                       REINTO+<256.*2>                 ;DATA
3163
3164
3165  012646  004037  034204                JSR     R0,@#CHREG      ;CHANGE BITS IN SAVED REGISTER
3166  012652  002222                       RHDS1                   ;CHANGE RHDS1 REGISTER
3167
3168  012654  000001                       1                       ;1 BIT/BITS TO BE CHANGED
3169  012656  000001                       1                       ;NEW VALUE OF LBT IS 1
3170  012660  002000                       LBT                     ;CHANGE LBT BIT
3171
3172  012662  004037  034204                JSR     R0,@#CHREG      ;CHANGE BITS IN SAVED REGISTER
3173  012666  002212                       RHCA                    ;CHANGE RHCA REGISTER
3174
3175  012670  000001                       1                       ;1 BIT/BITS TO BE CHANGED
3176  012672  000001                       1                       ;NEW VALUE OF BIT0 IS 1
3177  012674  000001                       BIT0                    ;CHANGE BIT0 BIT
3178
3179  012676  004037  033216                JSR     R0,@#FILLRE     ;MOV 0 INTO SAVED RHDST
3180  012702  002204                       RHDST                   ;SAVED REGISTER TO CHANGE
3181  012704  000000                       0                       ;DATA
3182
3183
3184                                        ;*COMPARE ALL REGISTERS
3185
3186  012706  004037  034312                JSR     R0,@#COMREG     ;COMPARE SAVED REGISTERS WITH
3187                                                                ;PRESENT VALUE
3188  012712  004512                       SAVERE                  ;GOOD DATA SAVED IN 'SAVERE'
3189  012714  002254                       WC                      ;TEST DATA STARTING FROM 'RHWC'
3190  012716  000022                       18.                     ;18. REGISTERS TO BE COMPARED
3191  012720  012724                       3$                      ;RETURN TO 3$ ON ERROR
3192  012722  012730                       4$                      ;RETURN TO 4$ ON NO ERROR
3193
3194
3195  012724  104045              3$:     ERROR   45              ;READING ON LAST BLOCK IE.
3196  012726  000207                       RTS     PC              ;CYLINDER 410./814., SECTOR 21, TRACK 18
3197                                                                ;CAUSED AN ERROR
3198                                                                ;GOOD DATA GIVES WHAT SHOULD
3199                                                                ;BE THERE
3200                                                                ;RECEIVED DATA GIVES WHAT
3201                                                                ;WAS THERE AFTER READ
3202                                                                ;FROM LAST BLOCK
3203
3204                                        ;*READ DATA WILL BE COMPARED
3205  012730                        4$:
3206
3207  012730  004037  035342                JSR     R0,@#COMPAR     ;COMPARE TWO BLOCKS OF MEMORY
3208  012734  002370                       WRFROM                  ;GOOD DATA STARTS FROM WRFROM
3209  012736  003434                       REINTO                  ;TEST DATA STARTS FROM REINTO
3210  012740  000400                       256.                    ;256., WORDS TO BE COMPARED
3211  012742  012746                       5$                      ;RETURN TO 5$ ON ERROR
3212  012744  012752                       6$                      ;RETURN TO 6$ ON NO ERROR
3213
3214
3215  012746  104046              5$:     ERROR   46              ;DATA READ FROM
```

```
 3216  012750  000207                        RTS     PC              ;LAST BLOCK IN ERROR
 3217
 3218  012752                          6$:
 3219
 3220
 3221
```

```
3222
3223                                    ;;*****************************************************************
3224                                    ;*TEST 12       SEARCH COMMAND
3225
3226                                    ;*      THE SEARCH COMMAND WILL BE DONE ON CYLINDER 0
3227                                    ;*      THAT IS STARTING WITH A RECALIBRATE
3228                                    ;*      THEN HEADER AND DATA WILL BE WRITTEN FOR SECTOR 0 AND 1
3229                                    ;*      ALL REGISTERS WILL BE CHECKED
3230                                    ;*      A SEARCH COMMAND WILL BE GIVEN FOR SECTOR 0
3231                                    ;*      ON INTERRUPT SECTOR 1 HEADER AND DATA WILL BE READ
3232                                    ;*      TIME WILL BE CRITICAL AS THE TIME TAKEN TO DO THE
3233                                    ;*      READ IS THE ONLY INDICATOR THAT THE HEADS WERE ON
3234                                    ;*      SECTOR 0 AT INTERRUPT TIME.  TIME ALLOWED IS MAXIMUM
3235                                    ;*      OF 1500 MICRO SECONDS
3236                                    ;*      THEN ALL REGISTERS ARE CHECKED AND DATA READ
3237                                    ;*      IS  CHECKED
3238
3239                                    ;;*****************************************************************
3240  012752  000004                    TST12:  SCOPE
3241  012754  012706  001000                    MOV     #STACK,SP       ;RESET STACK
3242  012760  012737  000012  004504            MOV     #12,@#TSTNM     ;SAVE TEST NUMBER
3243
3244  012766  004737  033314                    JSR     PC,@#CLDISK     ;SET R1-RHCS1, R2-RHCS2
3245                                                                    ;R3-RHDS1, R4-RHER1
3246                                                                    ;GIVE RH-11 INITIALIZE
3247                                                                    ;SETUP UNIT NUMBER
3248                                    ;*GET HEADS TO CYLINDER 0
3249  012772  004737  033374                    JSR     PC,@#CHECKT     ;CHECKS DVA,RDY,MOL,DPR,DRY AND VV = 1
3250                                                                    ;AND THAT ALL STATUS BITS ARE = 0
3251  012776  104401  057465                    TYPE    ,CPHALT         ;CANNOT CONTINUE TESTING IF NOT
3252  013002  000000                            HALT                    ;STOP TEST
3253
3254  013004  013777  004506  167154            MOV     @#RP4VEC,@RPVEC ;SET RP04 VECTOR ADDRESS
3255                                                                    ;TO 'TIME1' IF P-CLOCK IS PRESENT
3256                                                                    ;OR TO 'TIME2' IF P-CLOCK IS NOT THERE
3257                                                                    ;'TIME' WILL ONLY SAVE
3258                                                                    ;CURRENT CYLINDER ADDRESS
3259                                                                    ;AND LOOK AHEAD REGISTERS
3260
3261
3262  013012  013746  002326                    MOV     @#RECALI,-(SP)  ;GET READY TO MOVE COMMAND
3263  013016  052716  000101                    BIS     #GO!IE,(SP)     ;GET READY TO SET GO AND
3264                                                                    ;ENABLE INTERRUPT
3265  013022  012677  167152                    MOV     (SP)+,@RHCS1    ;GO WITH
3266                                                                     ;6 IN RHCS1 FOR RECALIBRATE
3267                                                                    ;WITH INTERRUPT ENABLED
3268
3269  013026  104413                            WAT                     ;WAIT FOR RDY BIT TO SET
3270  013030  002222                            RHDS1                   ;WAIT FOR RHDS1 REGISTER
3271  013032  000200                            RDY                     ;WAIT FOR RDY BIT IN RHDS1 REGISTER
3272  013034  012740                            5600.                   ;ALLOW 56000 MICRO SECONDS
3273  013036  012737                            5599.                   ;RDY MUST SET BETWEEN
3274                                                                    ;10 AND 111990 MICRO SECONDS
3275
3276
3277  013040  004737  033314                    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

013044  004037  033140               JSR     R0,@#FLHEAD     ;SAVE HEADER DATA IN WRFROM
013050  002370                       WRFROM                  ;LOCATION WHERE SAVED
013052  000004                       4                       ;NUMBER OF WORDS SAVED
013054  010000                       10000                   ;FIRST DATA WORD
013056  000000                       0                       ;SECOND DATA WORD
013060  000000                       0                       ;THIRD DATA WORD
013062  000000                       0                       ;FOURTH DATA WORD

                                    ;*FILL WRITE FROM BUFFER WITH DATA

013064  004037  033164               JSR     R0,@#CLAREA     ;CLEAR 256. WORDS, FROM WRFROM+10
013070  002400                       WRFROM+10               ;STARTING FROM WRFROM+10
013072  000400                       256.                    ;256. WORDS
013074  000000                       0                       ;FILL WITH 0


                                    ;*FILL WRITE FROM BUFFER WITH NEXT SECTOR HEADER

013076  004037  033140               JSR     R0,@#FLHEAD     ;SAVE HEADER DATA IN WRFROM+<260.*2>
013102  003400                       WRFROM+<260.*2>         ;LOCATION WHERE SAVED
013104  000004                       4                       ;NUMBER OF WORDS SAVED
013106  010000                       10000                   ;FIRST DATA WORD
013110  000001                       1                       ;SECOND DATA WORD
013112  000000                       0                       ;THIRD DATA WORD
013114  000000                       0                       ;FOURTH DATA WORD

                                    ;*FILL WRITE FROM BUFFER WITH NEXT SECTOR DATA

013116  004037  033164               JSR     R0,@#CLAREA     ;CLEAR 4 WORDS, FROM WRFROM+<264.*2>
013122  003410                       WRFROM+<264.*2>         ;STARTING FROM WRFROM+<264.*2>
013124  000004                       4                       ;4 WORDS
013126  000001                       1                       ;FILL WITH 1


                                    ;*CLEAR READ INTO BUFFER WITH DATA OTHER THAN EXPECTED DATA

013130  004037  033164               JSR     R0,@#CLAREA     ;CLEAR 260. WORDS, FROM REINTO
013134  003434                       REINTO                  ;STARTING FROM REINTO
013136  000404                       260.                    ;260. WORDS
013140  000377                       377                     ;FILL WITH 377


                                    ;*THE WRITE HEADER AND DATA WILL BE LOADED

013142  004037  035276               JSR     R0,@#RUN        ;SETUP TO RUN FOR DATA COMMAND
013146  000000                       0                       ;CYLINDER 0
013150     000                       .BYTE   0               ;SECTOR 0
013151     000                       .BYTE   0               ;TRACK 0
013152  177364                       -264.-4                 ;WORD COUNT (DATA) = 264. +
                                                            ;4 HEADER WORDS
```

```
3334  013154  002370                        WRFROM                  ;BUS ADDRESS
3335                                                                  ;STARTING ADDRESS OF DATA
3336                                                                  ;BUFFER = WRFROM
3337  013156  000000                        0                       ;DO NOT INHIBIT BUS ADDRESS INCREMENT
3338  013160  010000                        FMT22                   ;16 BITS PER WORD FORMAT
3339                                                                  ;DO NOT INHIBIT ECC CORRECTION
3340                                                                  ;DO NOT INHIBIT HEADER COMPARE
3341  013162  002344                        WRIFOR                  ;GET READY TO DO A WRIFOR
3342                                                        ;WRITE HEADER AND DATA WITH 62 IN RHCS1
3343
3344
3345                                        ;*SAVE REGISTERS FOR COMPARISON AFTER WRITE HEADER AND DATA
3346  013164  004037  033462                JSR     R0,@#SAVER      ;SAVE REGISTERS
3347  013170  002172                        RHWC                    ;RHWC IS THE FIRST REGISTER SAVED
3348  013172  004512                        SAVERE                  ;STARTING ADDRES OF WHERE
3349                                                                  ;THE REGISTERS ARE SAVED
3350  013174  000022                        18.                     ;NUMBER OF REGISTERS
3351                                                                  ;SAVED = 18.
3352
3353  013176  004737  033374                JSR     PC,@#CHECKT     ;CHECKS DVA,RDY,MOL,DPR,DRY AND VV = 1
3354                                                                  ;AND THAT ALL STATUS BITS ARE = 0
3355  013202  104401  057465                TYPE    ,CPHALT         ;CANNOT CONTINUE TESTING IF NOT
3356  013206  000000                        HALT                    ;STOP TEST
3357
3358  013210  013777  004506  166750        MOV     @#RP4VEC,@RPVEC ;SET RP04 VECTOR ADDRESS
3359                                                                  ;TO 'TIME1' IF P-CLOCK IS PRESENT
3360                                                                  ;OR TO 'TIME2' IF P-CLOCK IS NOT THERE
3361                                                                  ;'TIME' WILL ONLY SAVE
3362                                                                  ;CURRENT CYLINDER ADDRESS
3363                                                                  ;AND LOOK AHEAD REGISTERS
3364
3365
3366  013216  013746  002344                MOV     @#WRIFOR,-(SP)  ;GET READY TO MOVE COMMAND
3367  013222  052716  000101                BIS     #GO!IE,(SP)     ;GET READY TO SET GO AND
3368                                                                  ;ENABLE INTERRUPT
3369  013226  012677  166746                MOV     (SP)+,@RHCS1    ;GO WITH
3370                                                                   ;62 IN RHCS1 FOR WRITE HEADER AND DATA
3371                                                                  ;WITH INTERRUPT ENABLED
3372  013232  011100                        MOV     @R1,R0          ;SAVE RHCS1 DURING ABOVE OPERATION
3373  013234  011305                        MOV     @R3,R5          ;SAVE RHDS1 DURING ABOVE OPERATION
3374
3375                                        ;*ONE REVOLUTION=16670 MICRO SEC, ONE SECTOR = 760 MICRO SEC
3376
3377  013236  104413                        WAT                     ;WAIT FOR RDY BIT TO SET
3378  013240  002200                        RHCS1                   ;WAIT FOR RHCS1 REGISTER
3379  013242  000200                        RDY                     ;WAIT FOR RDY BIT IN RHCS1 REGISTER
3380  013244  001614                        908.                    ;ALLOW 9080 MICRO SECONDS
3381  013246  001507                        839.                    ;RDY MUST SET BETWEEN
3382                                                                  ;690 AND 17470 MICRO SECONDS
3383
3384                                        ;*COMPARE CONTENTS OF RHCS1 AND RHDS1 ALREADY SAVED IN
3385                                        ;*R0 AND R5 IMMEDIATELY AFTER GO IS ISSUED
3386
3387  013250  013746  002344                MOV     @#WRIFOR,-(SP)  ;SAVE COMMAND
3388  013254  052716  004101                BIS     #IE!DVA!GO,(SP) ;INCLUDE IE!DVA!GO
3389  013260  011637  001124                MOV     (SP),@#$GDDAT   ;SAVE FOR PRINTOUT
```

```
3390  013264  022600                          CMP     (SP)+,R0        ;DURING ABOVE OPERATION ONLY IE!DVA!GO
3391                                                                  ;AND COMMAND SHOULD BE SET
3392  013266  001405                          BEQ     67$             ;BRANCH IF GOOD
3393  013270  010037  001126                  MOV     R0,@#$BDDAT     ;BAD DATA
3394  013274  010137  004500                  MOV     R1,@#REGADR     ;FAILING REGISTER RHCS1
3395  013300  104021                          ERROR   21              ;DURING ABOVE OPERATION ONLY
3396                                                                  ;COMMAND AND IE!DVA!GO SHOULD BE SET
3397  013302  012746  010500          67$:    MOV     #MOL!DPR!VV,-(SP)       ;SAVE BITS SET DURING OPERATION IN RHDS1
3398  013306  011637  001124                  MOV     (SP),@#$GDDAT   ;SAVE FOR PRINTOUT
3399  013312  022605                          CMP     (SP)+,R5        ;DURING ABOVE OPERATION ONLY MOL!DPR!VV
3400                                                                  ;SHOULD BE SET
3401  013314  001405                          BEQ     69$             ;BRANCH IF GOOD
3402  013316  010537  001126                  MOV     R5,@#$BDDAT     ;BAD DATA
3403  013322  010337  004500                  MOV     R3,@#REGADR     ;FAILING REGISTER RHDS1
3404  013326  104063                          ERROR   63              ;DURING ABOVE OPERATION ONLY
3405                                                                  ;MOL!DPR!VV SHOULD BE SET
3406  013330                          69$:
3407
3408                                          ;*NOW CHANGE SAVE REGISTERS TO EXPECTED VALUES
3409
3410  013330  004037  033216                  JSR     R0,@#FILLRE     ;MOV 0 INTO SAVED RHWC
3411  013334  002172                          RHWC                    ;SAVED REGISTER TO CHANGE
3412  013336  000000                          0                       ;DATA
3413
3414
3415  013340  004037  033216                  JSR     R0,@#FILLRE     ;MOV WRFROM+<268.*2> INTO SAVED RHBA
3416  013344  002174                          RHBA                    ;SAVED REGISTER TO CHANGE
3417  013346  003420                          WRFROM+<268.*2>                 ;DATA
3418
3419
3420  013350  004037  033216                  JSR     R0,@#FILLRE     ;MOV 2 INTO SAVED RHDST
3421  013354  002204                          RHDST                   ;SAVED REGISTER TO CHANGE
3422  013356  000002                          2                       ;DATA
3423
3424
3425                                          ;*NOW COMPARE REGISTERS BEFORE WRITE HEADER AND DATA
3426                                          ;*WITH REGISTERS AFTER COMMAND
3427
3428  013360  004037  034312                  JSR     R0,@#COMREG     ;COMPARE SAVED REGISTERS WITH
3429                                                                  ;PRESENT VALUE
3430  013364  004512                          SAVERE                  ;GOOD DATA SAVED IN 'SAVERE'
3431  013366  002254                          WC                      ;TEST DATA STARTING FROM 'RHWC'
3432  013370  000022                          18.                     ;18. REGISTERS TO BE COMPARED
3433  013372  013376                          1$                      ;RETURN TO 1$ ON ERROR
3434  013374  013402                          2$                      ;RETURN TO 2$ ON NO ERROR
3435
3436
3437  013376  104027                  1$:     ERROR   27              ;WRITE HEADER AND DATA
3438  013400  000207                          RTS     PC              ;CAUSED IMPROPER REGISTER
3439                                                                  ;CHANGE
3440                                                                  ;GOOD DATA GIVES WHAT SHOULD
3441                                                                  ;BE THERE
3442                                                                  ;RECEIVED DATA GIVES WHAT
3443                                                                  ;WAS THERE AFTER COMMAND
3444
3445                                          ;*NOW A SEARCH COMMAND WILL BE GIVEN
```

```
                                        ;*BUT BEFORE THAT ALL POSSIBLE REGISTERS
                                        ;*WILL BE FILLED FOR THE READ HEADER AND DATA SECTOR 1
                                        ;*AS THERE WILL NOT BE MUCH TIME BETWEEN THE
                                        ;*COMPLETION OF THE SEARCH AND THE SECTOR 1 COMING.

                                        ;*FILL FOR THE READ HEADER AND DATA COMMAND WHICH WILL NOT
                                        ;*BE EXECUTED TILL AFTER THE SEARCH
                                        ;*THE SEARCH WILL ONLY LEAVE RHCS1 AND RHDST
                                        ;*CHANGED ALL THE REST WILL BE UNCHANGED

013402                          2$:

013402  004737  033314          JSR     PC,@#CLDISK     ;SET R1-RHCS1, R2-RHCS2
                                                        ;R3-RHDS1, R4-RHER1
                                                        ;GIVE RH-11 INITIALIZE
                                                        ;SETUP UNIT NUMBER

013406  004037  035276          JSR     R0,@#RUN        ;SETUP TO RUN FOR DATA COMMAND
013412  000000                  0                       ;CYLINDER 0
013414     000                  .BYTE   0               ;SECTOR 0
013415     000                  .BYTE   0               ;TRACK 0
013416  177770                  -8.                     ;WORD COUNT = 8.
013420  003434                  REINTO                  ;BUS ADDRESS
                                                        ;STARTING ADDRESS OF DATA
                                                        ;BUFFER = REINTO
013422  000000                  0                       ;DO NOT INHIBIT BUS ADDRESS INCREMENT
013424  014000                  ECI!FMT22                       ;16 BITS PER WORD FORMAT
                                                        ;INHIBIT ECC CORRECTION
                                                        ;DO NOT INHIBIT HEADER COMPARE
013426  002334                  SERCH                   ;GET READY TO DO A SERCH
                                                        ;SEARCH WITH 30 IN RHCS1


                                        ;*SAVE REGISTERS FOR COMPARISON NOT AFTER THE
                                        ;*SEARCH COMMAND BUT AFTER THE READ HEADER AND DATA
013430  004037  033462          JSR     R0,@#SAVER      ;SAVE REGISTERS
013434  002172                  RHWC                    ;RHWC IS THE FIRST REGISTER SAVED
013436  004512                  SAVERE                  ;STARTING ADDRES OF WHERE
                                                        ;THE REGISTERS ARE SAVED
013440  000022                  18.                     ;NUMBER OF REGISTERS
                                                        ;SAVED = 18.

                                        ;*NOW SAVE VALUES FOR RHCS1 AND RHDST WHICH
                                        ;*WILL CHANGE AFTER THE SEARCH

013442  013746  002350          MOV     @#REFOR,-(SP)   ;SAVE READ HEADER AND DATA
013446  052716  000101          BIS     #IE!GO,(SP)     ;INTERRUPT ENABLE AND GO
013452  012637  004652          MOV     (SP)+,@#TMP0    ;SAVE IN R0 FOR RHCS1
013456  012737  000001  004660  MOV     #1,@#TMP5       ;SAVE TRACK 0 SECTOR 1 FOR RHDST

                                        ;*THE INTERRUPT VECTOR WILL BE SET TO GO TO 2$
                                        ;*AFTER THE SEARCH

013464  012777  013532  166474  MOV     #7$,@RPVEC      ;SET INTERRUPT VECTOR TO 2$
013472  004737  033374          JSR     PC,@#CHECKT     ;CHECKS DVA,RDY,MOL,DPR,DRY AND VV = 1
                                                        ;AND THAT ALL STATUS BITS ARE = 0
```

```
3502  013476  104401  057465                TYPE    ,CPHALT          ;CANNOT CONTINUE TESTING IF NOT
3503  013502  000000                        HALT                     ;STOP TEST
3504
3505  013504  013746  002334                MOV     @#SERCH,-(SP)    ;GET READY TO MOVE COMMAND
3506  013510  052716  000101                BIS     #GO!IE,(SP)      ;GET READY TO SET GO AND
3507                                                                   ;ENABLE INTERRUPT
3508  013514  012677  166460                MOV     (SP)+,@RHCS1     ;GO WITH
3509                                                                   ;WITH INTERRUPT ENABLED
3510
3511                                        ;*TIME IS NOT CRITICAL THIS ONLY WAITS FOR SEARCH COMPLETION
3512
3513  013520  104413                        WAT                      ;WAIT FOR DRY BIT TO SET
3514  013522  002222                        RHDS1                    ;WAIT FOR RHDS1 REGISTER
3515  013524  000200                        DRY                      ;WAIT FOR DRY BIT IN RHDS1 REGISTER
3516  013526  001614                        908.                     ;ALLOW 9080 MICRO SECONDS
3517  013530  001507                        839.                     ;DRY MUST SET BETWEEN
3518                                                                   ;690 AND 17470 MICRO SECONDS
3519
3520  013532  012737  000000  177776  7$:   MOV     #0,PS            ;SET PROSESSOR STATUS TO
3521                                                                   ;PRIORITY 0 IN CASE IT WAS
3522                                                                   ;TAKEN OUT OF WAT ROUTINE
3523                                                                   ;BEFORE RTI
3524  013540  013777  004660  166436          MOV     @#TMP5,@RHDST    ;SET DESIRED SECTOR/TRACK
3525                                                                   ;REGISTER TO SECTOR 1,TRACK 0
3526  013546  013777  004506  166412          MOV     @#RP4VEC,@RPVEC  ;SET RP04 VECTOR ADDRESS
3527                                                                   ;TO 'TIME1' IF P-CLOCK IS PRESENT
3528                                                                   ;OR TO 'TIME2' IF P-CLOCK IS NOT THERE
3529                                                                   ;'TIME' WILL ONLY SAVE
3530                                                                   ;CURRENT CYLINDER ADDRESS
3531                                                                   ;AND LOOK AHEAD REGISTERS
3532
3533  013554  013777  004652  166416          MOV     @#TMP0,@RHCS1    ;FILL RHCS1 WITH READ COMMAND
3534                                                                   ;TOGETHER WITH INTERRUPT ENABLE
3535                                                                   ;AND GO
3536
3537                                        ;*TIME ALLOWED HERE IS CRITICAL ANY TIME ERROR
3538                                        ;*INDICATES WRONG SEARCH IN THE SEARCH COMMAND
3539
3540  013562  104413                        WAT                      ;WAIT FOR RDY BIT TO SET
3541  013564  002200                        RHCS1                    ;WAIT FOR RHCS1 REGISTER
3542  013566  000200                        RDY                      ;WAIT FOR RDY BIT IN RHCS1 REGISTER
3543  013570  000225                        149.                     ;ALLOW 1490 MICRO SECONDS
3544  013572  000002                        2.                       ;RDY MUST SET BETWEEN
3545                                                                   ;1470 AND 1510 MICRO SECONDS
3546
3547                                        ;*WRITE FROM BUFFER WILL BE FILLED WITH EXPECTED DATA
3548
3549  013574  004037  033140                JSR     R0,@#FLHEAD      ;SAVE HEADER DATA IN WRFROM
3550  013600  002370                        WRFROM                   ;LOCATION WHERE SAVED
3551  013602  000004                        4                        ;NUMBER OF WORDS SAVED
3552  013604  010000                        10000                    ;FIRST DATA WORD
3553  013606  000001                        1                        ;SECOND DATA WORD
3554  013610  000000                        0                        ;THIRD DATA WORD
3555  013612  000000                        0                        ;FOURTH DATA WORD
3556
3557  013614  004037  033164                JSR     R0,@#CLAREA      ;CLEAR 4 WORDS, FROM WRFROM+<4*2>
```

```
 3558  013620  002400                         WRFROM+<4*2>                    ;STARTING FROM WRFROM+<4*2>
 3559  013622  000004                         4                       ;4 WORDS
 3560  013624  000001                         1                       ;FILL WITH 1
 3561
 3562
 3563                                         ;*CHANGE SAVED REGISTERS TO EXPECTED VALUES
 3564
 3565  013626  004037  033216                 JSR     R0,@#FILLRE     ;MOV 0 INTO SAVED RHWC
 3566  013632  002172                         RHWC                    ;SAVED REGISTER TO CHANGE
 3567  013634  000000                         0                       ;DATA
 3568
 3569
 3570  013636  004037  033216                 JSR     R0,@#FILLRE     ;MOV REINTO+<8.*2> INTO SAVED RHBA
 3571  013642  002174                         RHBA                    ;SAVED REGISTER TO CHANGE
 3572  013644  003454                         REINTO+<8.*2>                   ;DATA
 3573
 3574
 3575  013646  004037  033216                 JSR     R0,@#FILLRE     ;MOV 4272 INTO SAVED RHCS1
 3576  013652  002200                         RHCS1                   ;SAVED REGISTER TO CHANGE
 3577  013654  004272                         4272                    ;DATA
 3578
 3579
 3580  013656  004037  033216                 JSR     R0,@#FILLRE     ;MOV 2 INTO SAVED RHDST
 3581  013662  002204                         RHDST                   ;SAVED REGISTER TO CHANGE
 3582  013664  000002                         2                       ;DATA
 3583
 3584
 3585                                         ;*COMPARE REGISTER BEFORE READ HEADER AND DATA
 3586                                         ;*WITH REGISTERS AFTER COMMAND
 3587
 3588
 3589  013666  004037  034312                 JSR     R0,@#COMREG     ;COMPARE SAVED REGISTERS WITH
 3590                                                                 ;PRESENT VALUE
 3591  013672  004512                         SAVERE                  ;GOOD DATA SAVED IN 'SAVERE'
 3592  013674  002254                         WC                      ;TEST DATA STARTING FROM 'RHWC'
 3593  013676  000022                         18.                     ;18. REGISTERS TO BE COMPARED
 3594  013700  013704                         3$                      ;RETURN TO 3$ ON ERROR
 3595  013702  013710                         4$                      ;RETURN TO 4$ ON NO ERROR
 3596
 3597  013704  104031          3$:            ERROR   31              ;READ HEADER AND DATA CAUSED
 3598  013706  000207                         RTS     PC              ;IMPROPER REGISTER CHANGE
 3599                                                                 ;GOOD DATA GIVES WHAT SHOULD
 3600                                                                 ;BE THERE
 3601                                                                 ;RECEIVED DATA GIVES WHAT WAS
 3602                                                                 ;THERE AFTER COMMAND
 3603
 3604                                         ;*NOW READ INTO BUFFER WILL BE CHECKED TO SEE
 3605                                         ;*THE READ WAS GOOD
 3606  013710          4$:
 3607
 3608  013710  004037  035342                 JSR     R0,@#COMPAR     ;COMPARE TWO BLOCKS OF MEMORY
 3609  013714  002370                         WRFROM                  ;GOOD DATA STARTS FROM WRFROM
 3610  013716  003434                         REINTO                  ;TEST DATA STARTS FROM REINTO
 3611  013720  000010                         8.                      ;8., WORDS TO BE COMPARED
 3612  013722  013726                         5$                      ;RETURN TO 5$ ON ERROR
 3613  013724  013732                         6$                      ;RETURN TO 6$ ON NO ERROR
```

```
 3614
 3615
 3616  013726  104053                5$:     ERROR   53              ;READ HEADER AND DATA
 3617  013730  000207                        RTS     PC              ;AFTER A SEARCH CAUSED
 3618                                                                ;AN ERROR
 3619  013732                        6$:
 3620
```

```
3621
3622                                    ;;*****************************************************************
3623                                    ;*TEST 13       SEARCH COMMAND
3624
3625                                    ;*      THE ONLY THING NEW IN THIS TEST IS AN IMPLIED SEEK
3626                                    ;*      IN A SEARCH COMMAND
3627                                    ;*      THE HEADS START FROM CYLINDER 10 BY A SEEK
3628                                    ;*      COMMAND THEN A SEARCH SECTOR 0 TRACK 0 CYLINDER 0
3629                                    ;*      IS GIVEN
3630                                    ;*      THEN A READ COMMAND IS GIVEN FOR
3631                                    ;*      CYLINDER 0, TRACK 0, SECTOR 1
3632                                    ;*      TIME FOR THE READ IS THE ONLY INDICATOR
3633                                    ;*      OF CORRECT SEARCH
3634
3635                                    ;;*****************************************************************
3636  013732  000004                   TST13:  SCOPE
3637  013734  012706  001000                   MOV     #STACK,SP       ;RESET STACK
3638  013740  012737  000013  004504           MOV     #13,@#TSTNM     ;SAVE TEST NUMBER
3639
3640  013746  004737  033314                   JSR     PC,@#CLDISK     ;SET R1-RHCS1, R2-RHCS2
3641                                                                    ;R3-RHDS1, R4-RHER1
3642                                                                    ;GIVE RH-11 INITIALIZE
3643                                                                    ;SETUP UNIT NUMBER
3644
3645                                           ;*GET THE HEADS TO CYLINDER 10
3646  013752  004737  033374                   JSR     PC,@#CHECKT     ;CHECKS DVA,RDY,MOL,DPR,DRY AND VV = 1
3647                                                                    ;AND THAT ALL STATUS BITS ARE = 0
3648  013756  104401  057465                   TYPE    ,CPHALT         ;CANNOT CONTINUE TESTING IF NOT
3649  013762  000000                           HALT                    ;STOP TEST
3650  013764  013777  004506  166174           MOV     @#RP4VEC,@RPVEC ;SET RP04 VECTOR ADDRESS
3651                                                                    ;TO 'TIME1' IF P-CLOCK IS PRESENT
3652                                                                    ;OR TO 'TIME2' IF P-CLOCK IS NOT THERE
3653                                                                    ;'TIME' WILL ONLY SAVE
3654                                                                    ;CURRENT CYLINDER ADDRESS
3655                                                                    ;AND LOOK AHEAD REGISTERS
3656
3657  013772  004037  033264                   JSR     R0,@#SEEKCY     ;SEEK FOR
3658  013776  000012                           10.                     ;CYLINDER 10.
3659
3660
3661  014000  013746  002352                   MOV     @#SEECOM,-(SP)  ;GET READY TO MOVE COMMAND
3662  014004  052716  000101                   BIS     #GO!IE,(SP)     ;GET READY TO SET GO AND
3663                                                                    ;ENABLE INTERRUPT
3664  014010  012677  166164                   MOV     (SP)+,@RHCS1    ;GO WITH
3665                                                                     ;4 IN RHCS1 FOR SEEK
3666                                                                    ;WITH INTERRUPT ENABLED
3667
3668  014014  104413                           WAT                     ;WAIT FOR DRY BIT TO SET
3669  014016  002222                           RHDS1                   ;WAIT FOR RHDS1 REGISTER
3670  014020  000200                           DRY                     ;WAIT FOR DRY BIT IN RHDS1 REGISTER
3671  014022  015530                           7000.                   ;ALLOW 70000 MICRO SECONDS
3672  014024  000043                           35.                     ;DRY MUST SET BETWEEN
3673                                                                    ;69650 AND 70350 MICRO SECONDS
3674
3675                                          ;*FILL REGISTERS FOR READ HEADER AND DATA TO BE DONE AFTER SEARCH
3676
```

```
3677  014026  004737  033314                  JSR     PC,@#CLDISK     ;SET R1-RHCS1, R2-RHCS2
3678                                                                    ;R3-RHDS1, R4-RHER1
3679                                                                    ;GIVE RH-11 INITIALIZE
3680                                                                    ;SETUP UNIT NUMBER
3681
3682  014032  004037  035276                  JSR     R0,@#RUN        ;SETUP TO RUN FOR DATA COMMAND
3683  014036  000000                          0                       ;CYLINDER 0
3684  014040     000                          .BYTE   0               ;SECTOR 0
3685  014041     000                          .BYTE   0               ;TRACK 0
3686  014042  177770                          -8.                     ;WORD COUNT = 8.
3687  014044  003434                          REINTO                  ;BUS ADDRESS
3688                                                                    ;STARTING ADDRESS OF DATA
3689                                                                    ;BUFFER = REINTO
3690  014046  000000                          0                       ;DO NOT INHIBIT BUS ADDRESS INCREMENT
3691  014050  014000                          EC1!FMT22                       ;16 BITS PER WORD FORMAT
3692                                                                    ;INHIBIT ECC CORRECTION
3693                                                                    ;DO NOT INHIBIT HEADER COMPARE
3694  014052  002334                          SERCH                   ;GET READY TO DO A SERCH
3695                                                                    ;SEARCH WITH 30 IN RHCS1
3696
3697
3698                                          ;*SAVE REGISTERS FOR COMPARISON AFTER SEARCH
3699                                          ;*AND READ HEADER AND DATA
3700  014054  004037  033462                  JSR     R0,@#SAVER      ;SAVE REGISTERS
3701  014060  002172                          RHWC                    ;RHWC IS THE FIRST REGISTER SAVED
3702  014062  004512                          SAVERE                  ;STARTING ADDRES OF WHERE
3703                                                                    ;THE REGISTERS ARE SAVED
3704  014064  000022                          18.                     ;NUMBER OF REGISTERS
3705                                                                    ;SAVED = 18.
3706
3707                                          ;*NOW GIVE THE SEARCH COMMAND
3708  014066  004737  033374                  JSR     PC,@#CHECKT     ;CHECKS DVA,RDY,MOL,DPR,DRY AND VV = 1
3709                                                                    ;AND THAT ALL STATUS BITS ARE = 0
3710  014072  104401  057465                  TYPE    ,CPHALT         ;CANNOT CONTINUE TESTING IF NOT
3711  014076  000000                          HALT                    ;STOP TEST
3712  014100  012777  014264  166060          MOV     #3$,@RPVEC      ;INTERRUPT VECTOR SET TO 3$
3713
3714  014106  004037  033244                  JSR     R0,@#SRCH       ;SEARCH FOR
3715  014112  000000                          0                       ;CYLINDER 0
3716  014114     000                  .BYTE   0                       ;SECTOR 0
3717  014115     000                  .BYTE   0                       ;TRACK 0
3718
3719  014116  013700  002334                  MOV     @#SERCH,R0      ;EXPECTED CONTENTS OF RHCS1
3720                                                                    ;IMMEDIATELY AFTER GO
3721  014122  052700  004301                  BIS     #DVA!RDY!IE!GO,R0 ;EXPECTED BITS IN RHCS1
3722  014126  012705  010500                  MOV     #MOL!DPR!VV,R5  ;EXPECTED BITS IN RHDS1
3723                                                                    ;IMMEDIATELY AFTER GO
3724
3725
3726  014132  013746  002334                  MOV     @#SERCH,-(SP)   ;GET READY TO MOVE COMMAND
3727  014136  052716  000101                  BIS     #GO!IE,(SP)     ;GET READY TO SET GO AND
3728                                                                    ;ENABLE INTERRUPT
3729  014142  012677  166032                  MOV     (SP)+,@RHCS1    ;GO WITH
3730                                                                    ;WITH INTERRUPT ENABLED
3731  014146  021100                          CMP     @R1,R0          ;IS RHCS1 GOOD
3732  014150  001413                          BEQ     1$              ;BRANCH IF GOOD
```

```
3733  014152  011137  001126                  MOV     @R1,@#$BDDAT    ;BAD DATA FOR RHCS1
3734  014156  010037  001124                  MOV     R0,@#$GDDAT     ;GOOD DATA
3735  014162  010137  004500                  MOV     R1,@#REGADR     ;FAILING REGISTER RHCS1
3736  014166  012737  000340  000036          MOV     #340,@#TRAPVEC+2 ;TRAP PRIORITY = 7
3737  014174  104021                          ERROR   21              ;DURING SEARCH COMMAND
3738                                                                  ;CONTENTS OF RHCS1 WAS
3739                                                                  ;NOT AS EXPECTED
3740  014176  000414                          BR      2$              ;IF LAST ERROR 21 OCCURRED
3741                                                                  ;THEN DO NOT CHECK RHDS1
3742                                                                  ;AS TOO MUCH TIME HAS
3743                                                                  ;PASSED
3744
3745  014200  021305                  1$:     CMP     @R3,R5          ;IS RHDS1 GOOD
3746  014202  001412                          BEQ     2$              ;BRANCH IF GOOD
3747  014204  011337  001126                  MOV     @R3,@#$BDDAT    ;BAD DATA FOR RHDS1
3748  014210  010537  001124                  MOV     R5,@#$GDDAT     ;GOOD DATA
3749  014214  010337  004500                  MOV     R3,@#REGADR     ;FAILING REGISTER RHDS1
3750  014220  012737  000340  000036          MOV     #340,@#TRAPVEC+2 ;TRAP PRIORITY = 7
3751  014226  104063                          ERROR   63              ;DURING SEARCH COMMAND
3752                                                                  ;CONTENTS OF RHDS1 WAS
3753                                                                  ;IN CORRECT
3754
3755  014230  013737  002350  004652  2$:     MOV     @#REFOR,@#TMP0  ;SAVE READ HEADER AND DATA
3756  014236  052737  000101  004652          BIS     #IE!GO,@#TMP0   ;INCLUDE INTERRUPT ENABLE, GO
3757  014244  012737  000001  004660          MOV     #1,@#TMP5       ;SAVE TRACK 0, SECTOR 1
3758
3759                                          ;*THIS IS ONLY A WAIT LOOP
3760
3761  014252  104413                          WAT                     ;WAIT FOR RDY BIT TO SET
3762  014254  002222                          RHDS1                   ;WAIT FOR RHDS1 REGISTER
3763  014256  000200                          RDY                     ;WAIT FOR RDY BIT IN RHDS1 REGISTER
3764  014260  015530                          7000.                   ;ALLOW 70000 MICRO SECONDS
3765  014262  000043                          35.                     ;RDY MUST SET BETWEEN
3766                                                                  ;69650 AND 70350 MICRO SECONDS
3767
3768  014264  012737  000200  000036  3$:     MOV     #200,@#TRAPVEC+2 ;TRAP PRIORITY = 4
3769  014272  012737  000000  177776          MOV     #0,PS           ;SET PROSESSOR STATUS TO 0
3770  014300  013777  004660  165676          MOV     @#TMP5,@RHDST   ;SET DESIRED SECTOR/TRACK
3771                                                                  ;REGISTER TO SECTOR 1, TRACK 0
3772
3773  014306  013777  004506  165652          MOV     @#RP4VEC,@RPVEC ;SET RP04 VECTOR ADDRESS
3774                                                                  ;TO 'TIME1' IF P-CLOCK IS PRESENT
3775                                                                  ;OR TO 'TIME2' IF P-CLOCK IS NOT THERE
3776                                                                  ;'TIME' WILL ONLY SAVE
3777                                                                  ;CURRENT CYLINDER ADDRESS
3778                                                                  ;AND LOOK AHEAD REGISTERS
3779  014314  013711  004652                  MOV     @#TMP0,@R1      ;FILL RHCS1 WITH READ COMMAND
3780                                                                  ;TOGETHER WITH INTERRUPT ENABLE
3781                                                                  ;AND GO
3782
3783                                          ;*TIME ALLOWED HERE IS CRITICAL ANY TIME ERROR
3784                                          ;*INDICATES WRONG SEARCH IN THE SEARCH COMMAND
3785
3786  014320  104413                          WAT                     ;WAIT FOR RDY BIT TO SET
3787  014322  002200                          RHCS1                   ;WAIT FOR RHCS1 REGISTER
3788  014324  000200                          RDY                     ;WAIT FOR RDY BIT IN RHCS1 REGISTER
```

```
 3789  014326  000225                        149.                    ;ALLOW 1490 MICRO SECONDS
 3790  014330  000002                        2.                      ;RDY MUST SET BETWEEN
 3791                                                                  ;1470 AND 1510 MICRO SECONDS
 3792
 3793                                        ;*WRITE FROM BUFFER WILL BE FILLED WITH EXPECTED DATA
 3794
 3795  014332  004037  033140                JSR     R0,@#FLHEAD     ;SAVE HEADER DATA IN WRFROM
 3796  014336  002370                        WRFROM                  ;LOCATION WHERE SAVED
 3797  014340  000004                        4                       ;NUMBER OF WORDS SAVED
 3798  014342  010000                        10000                   ;FIRST DATA WORD
 3799  014344  000001                        1                       ;SECOND DATA WORD
 3800  014346  000000                        0                       ;THIRD DATA WORD
 3801  014350  000000                        0                       ;FOURTH DATA WORD
 3802
 3803  014352  004037  033164                JSR     R0,@#CLAREA     ;CLEAR 4 WORDS, FROM WRFROM+<4*2>
 3804  014356  002400                        WRFROM+<4*2>                    ;STARTING FROM WRFROM+<4*2>
 3805  014360  000004                        4                       ;4 WORDS
 3806  014362  000001                        1                       ;FILL WITH 1
 3807
 3808
 3809                                        ;*CHANGE SAVED REGISTERS TO EXPECTED VALUES
 3810
 3811  014364  004037  033216                JSR     R0,@#FILLRE     ;MOV 0 INTO SAVED RHWC
 3812  014370  002172                        RHWC                    ;SAVED REGISTER TO CHANGE
 3813  014372  000000                        0                       ;DATA
 3814
 3815
 3816  014374  004037  033216                JSR     R0,@#FILLRE     ;MOV REINTO+<8.*2> INTO SAVED RHBA
 3817  014400  002174                        RHBA                    ;SAVED REGISTER TO CHANGE
 3818  014402  003454                        REINTO+<8.*2>                   ;DATA
 3819
 3820
 3821  014404  004037  033216                JSR     R0,@#FILLRE     ;MOV 4272 INTO SAVED RHCS1
 3822  014410  002200                        RHCS1                   ;SAVED REGISTER TO CHANGE
 3823  014412  004272                        4272                    ;DATA
 3824
 3825
 3826  014414  004037  033216                JSR     R0,@#FILLRE     ;MOV 2 INTO SAVED RHDST
 3827  014420  002204                        RHDST                   ;SAVED REGISTER TO CHANGE
 3828  014422  000002                        2                       ;DATA
 3829
 3830
 3831  014424  004037  033216                JSR     R0,@#FILLRE     ;MOV 0 INTO SAVED RHCC
 3832  014430  002234                        RHCC                    ;SAVED REGISTER TO CHANGE
 3833  014432  000000                        0                       ;DATA
 3834
 3835
 3836                                        ;*COMPARE REGISTER BEFORE READ HEADER AND DATA
 3837                                        ;*WITH REGISTERS AFTER COMMAND
 3838
 3839
 3840  014434  004037  034312                JSR     R0,@#COMREG     ;COMPARE SAVED REGISTERS WITH
 3841                                                                  ;PRESENT VALUE
 3842  014440  004512                        SAVERE                  ;GOOD DATA SAVED IN 'SAVERE'
 3843  014442  002254                        WC                      ;TEST DATA STARTING FROM 'RHWC'
 3844  014444  000022                        18.                     ;18. REGISTERS TO BE COMPARED
```

```
3845  014446  014452                    4$                      ;RETURN TO 4$ ON ERROR
3846  014450  014456                    5$                      ;RETURN TO 5$ ON NO ERROR
3847
3848
3849  014452  104031              4$:   ERROR   31              ;READ HEADER AND DATA CAUSED
3850  014454  000207                    RTS     PC              ;IMPROPER REGISTER CHANGE
3851                                                            ;GOOD DATA GIVES WHAT SHOULD
3852                                                            ;BE THERE
3853                                                            ;RECEIVED DATA GIVES WHAT WAS
3854                                                            ;THERE AFTER COMMAND
3855
3856                                    ;*NOW READ INTO BUFFER WILL BE CHECKED TO SEE
3857                                    ;*THE READ WAS GOOD
3858  014456                    5$:
3859
3860  014456  004037  035342           JSR     R0,@#COMPAR     ;COMPARE TWO BLOCKS OF MEMORY
3861  014462  002370                    WRFROM                  ;GOOD DATA STARTS FROM WRFROM
3862  014464  003434                    REINTO                  ;TEST DATA STARTS FROM REINTO
3863  014466  000010                    8.                      ;8., WORDS TO BE COMPARED
3864  014470  014474                    6$                      ;RETURN TO 6$ ON ERROR
3865  014472  014500                    7$                      ;RETURN TO 7$ ON NO ERROR
3866
3867
3868  014474  104053              6$:   ERROR   53              ;READ HEADER AND DATA
3869  014476  000207                    RTS     PC              ;AFTER A SEARCH CAUSED
3870                                                            ;AN ERROR
3871  014500                    7$:
3872
3873
3874
3875
```

```
3876
3877                                    ;;*****************************************************************
3878                                    ;*      THE NEXT TEST REMOVES SECTOR 1 ON CYLINDER 0
3879                                    ;*      TRACK0 AND PUTS SECTOR 0 THERE.
3880                                    ;*      HENCE THE PACK IS UNFORMATTED FROM
3881                                    ;*      THIS POINT ON TO THE TEST WHEN SECTOR
3882                                    ;*      1 IS REPLACED.  IF TESTING IS STOPPED WITH
3883                                    ;*      AN ERROR IN THE SECTION OF THE PROGRAM BETWEEN
3884                                    ;*      THIS AND WHEN SECTOR 1 IS REPLACED THEN THE
3885                                    ;*      DISK BEING USED MAY HAVE BEEN UNFORMATTED
3886                                    ;*      IF THE LAST PASS OF THIS PROGRAM GIVES
3887                                    ;*      NO ERRORS IN THIS SECTION THEN THE DISK
3888                                    ;*      MAY NOT HAVE BEEN UNFORMATTED. HOWEVER IT
3889                                    ;*      IS RECOMMENDED THAT AFTER A PASS OF THIS
3890                                    ;*      PROGRAM THE DISK BE REFORMATTED.
3891                                    ;;*****************************************************************
3892
3893
3894
3895
3896
3897
3898                                    ;;*****************************************************************
3899                                    ;*TEST 14       HEADER COMPARE ERROR - RHER1 BIT #7 (HCE)
3900
3901                                    ;*      WRITE HEADER AND DATA IS USED TO REMOVE SECTOR 1
3902                                    ;*      AND PUT SECTOR 0 THERE ON CYLINDER 0
3903                                    ;*      THEN A READ DATA IS GIVEN FOR SECTOR1
3904                                    ;*      HCE- BIT #7 IN RHER1 SHOULD SET.
3905                                    ;*      ALL REGISTERS ARE CHECKED
3906                                    ;*      ANY DATA READ IS CHECKED
3907
3908                                    ;;*****************************************************************
3909  014500  000004                   TST14:  SCOPE
3910  014502  012706  001000                   MOV     #STACK,SP       ;RESET STACK
3911  014506  012737  000014  004504           MOV     #14,@#TSTNM     ;SAVE TEST NUMBER
3912
3913  014514  004737  033314                   JSR     PC,@#CLDISK     ;SET R1-RHCS1, R2-RHCS2
3914                                                                    ;R3-RHDS1, R4-RHER1
3915                                                                    ;GIVE RH-11 INITIALIZE
3916                                                                    ;SETUP UNIT NUMBER
3917
3918                                           ;*FILL WRITE FROM BUFFER WITH HEADER
3919
3920  014520  004037  033140                   JSR     R0,@#FLHEAD     ;SAVE HEADER DATA IN WRFROM
3921  014524  002370                           WRFROM                  ;LOCATION WHERE SAVED
3922  014526  000005                           5                       ;NUMBER OF WORDS SAVED
3923  014530  010000                           10000                   ;FIRST DATA WORD
3924  014532  000000                           0                       ;SECOND DATA WORD
3925  014534  000000                           0                       ;THIRD DATA WORD
3926  014536  000000                           0                       ;FOURTH DATA WORD
3927  014540  000001                           1                       ;FIFTH DATA WORD
3928
3929                                           ;*FILL READ INTO BUFFER WITH ALL ONES
3930
3931
```

```
3932  014542  004037  033164                  JSR     R0,@#CLAREA     ;CLEAR 256. WORDS, FROM REINTO
3933  014546  003434                          REINTO                  ;STARTING FROM REINTO
3934  014550  000400                          256.                    ;256. WORDS
3935  014552  177777                          -1                      ;FILL WITH -1
3936
3937
3938                                          ;*WRITE HEADER AND DATA IS LOADED
3939
3940  014554  004037  035276                  JSR     R0,@#RUN        ;SETUP TO RUN FOR DATA COMMAND
3941  014560  000000                          0.                      ;CYLINDER 0.
3942  014562     001                          .BYTE   1.              ;SECTOR 1.
3943  014563     000                          .BYTE   0.              ;TRACK 0.
3944  014564  177773                          -1-4                    ;WORD COUNT (DATA) = 1 +
3945                                                                  ;4 HEADER WORDS
3946  014566  002370                          WRFROM                  ;BUS ADDRESS
3947                                                                  ;STARTING ADDRESS OF DATA
3948                                                                  ;BUFFER = WRFROM
3949  014570  000000                          0                       ;DO NOT INHIBIT BUS ADDRESS INCREMENT
3950  014572  010000                          FMT22                   ;16 BITS PER WORD FORMAT
3951                                                                  ;DO NOT INHIBIT ECC CORRECTION
3952                                                                  ;DO NOT INHIBIT HEADER COMPARE
3953  014574  002344                          WRIFOR                  ;GET READY TO DO A WRIFOR
3954                                                  ;WRITE HEADER AND DATA WITH 62 IN RHCS1
3955
3956
3957                                          ;*NOW SAVE REGISTERS FOR COMPARISON AFTER WRITE
3958
3959  014576  004037  033462                  JSR     R0,@#SAVER      ;SAVE REGISTERS
3960  014602  002172                          RHWC                    ;RHWC IS THE FIRST REGISTER SAVED
3961  014604  004512                          SAVERE                  ;STARTING ADDRES OF WHERE
3962                                                                  ;THE REGISTERS ARE SAVED
3963  014606  000021                          17.                     ;NUMBER OF REGISTERS
3964                                                                  ;SAVED = 17.
3965
3966  014610  004737  033374                  JSR     PC,@#CHECKT     ;CHECKS DVA,RDY,MOL,DPR,DRY AND VV = 1
3967                                                                  ;AND THAT ALL STATUS BITS ARE = 0
3968  014614  104401  057465                  TYPE    ,CPHALT         ;CANNOT CONTINUE TESTING IF NOT
3969  014620  000000                          HALT                    ;STOP TEST
3970
3971  014622  013777  004506  165336          MOV     @#RP4VEC,@RPVEC ;SET RP04 VECTOR ADDRESS
3972                                                                  ;TO 'TIME1' IF P-CLOCK IS PRESENT
3973                                                                  ;OR TO 'TIME2' IF P-CLOCK IS NOT THERE
3974                                                                  ;'TIME' WILL ONLY SAVE
3975                                                                  ;CURRENT CYLINDER ADDRESS
3976                                                                  ;AND LOOK AHEAD REGISTERS
3977
3978
3979  014630  013746  002344                  MOV     @#WRIFOR,-(SP)  ;GET READY TO MOVE COMMAND
3980  014634  052716  000101                  BIS     #GO!IE,(SP)     ;GET READY TO SET GO AND
3981                                                                  ;ENABLE INTERRUPT
3982  014640  012677  165334                  MOV     (SP)+,@RHCS1    ;GO WITH
3983                                                                   ;62 IN RHCS1 FOR WRITE HEADER AND DATA
3984                                                                  ;WITH INTERRUPT ENABLED
3985
3986                                          ;*TIME IS NOT CRITICAL
3987
```

```
3988  014644  104413                        WAT                     ;WAIT FOR RDY BIT TO SET
3989  014646  002200                        RHCS1                   ;WAIT FOR RHCS1 REGISTER
3990  014650  000200                        RDY                     ;WAIT FOR RDY BIT IN RHCS1 REGISTER
3991  014652  004704                        2500.                   ;ALLOW 25000 MICRO SECONDS
3992  014654  004704                        2500.                   ;RDY MUST SET BETWEEN
3993                                                                ;00 AND 50000 MICRO SECONDS
3994
3995                                        ;*NOW CHANGE SAVED REGISTERS TO EXPECTED VALUE
3996
3997  014656  004037  033216                JSR     R0,@#FILLRE     ;MOV 0 INTO SAVED RHWC
3998  014662  002172                        RHWC                    ;SAVED REGISTER TO CHANGE
3999  014664  000000                        0                       ;DATA
4000
4001
4002  014666  004037  033216                JSR     R0,@#FILLRE     ;MOV WRFROM+<5*2> INTO SAVED RHBA
4003  014672  002174                        RHBA                    ;SAVED REGISTER TO CHANGE
4004  014674  002402                        WRFROM+<5*2>                    ;DATA
4005
4006
4007  014676  004037  033216                JSR     R0,@#FILLRE     ;MOV 2 INTO SAVED RHDST
4008  014702  002204                        RHDST                   ;SAVED REGISTER TO CHANGE
4009  014704  000002                        2                       ;DATA
4010
4011
4012                                        ;*COMPARE ALL REGISTERS
4013
4014
4015  014706  004037  034312                JSR     R0,@#COMREG     ;COMPARE SAVED REGISTERS WITH
4016                                                                ;PRESENT VALUE
4017  014712  004512                        SAVERE                  ;GOOD DATA SAVED IN 'SAVERE'
4018  014714  002254                        WC                      ;TEST DATA STARTING FROM 'RHWC'
4019  014716  000021                        17.                     ;17. REGISTERS TO BE COMPARED
4020  014720  014724                        1$                      ;RETURN TO 1$ ON ERROR
4021  014722  014730                        2$                      ;RETURN TO 2$ ON NO ERROR
4022
4023  014724  104027              1$:       ERROR   27              ;WRITING HEADER AND DATA CAUSED
4024  014726  000207                        RTS     PC
4025                                                                ;IMPROPER REGISTER CHANGE
4026                                                                ;GOOD DATA GIVES WHAT
4027                                                                ;SHOULD BE THERE
4028                                                                ;RECEIVED DATA GIVES WHAT
4029                                                                ;WAS THERE AFTER WRITE
```

```
4030
4031
4032                                        ;*NOW A READ DATA WILL BE DONE ON CYLINDER=0, SECTOR=1,
4033                                        ;*TRACK=0
4034                                        ;*FILL WRITE FROM BUFFER WITH EXPECTED DATA
4035
4036  014730                          2$:
4037
4038  014730  004737  033314               JSR     PC,@#CLDISK     ;SET R1-RHCS1, R2-RHCS2
4039                                                                 ;R3-RHDS1, R4-RHER1
4040                                                                 ;GIVE RH-11 INITIALIZE
4041                                                                 ;SETUP UNIT NUMBER
4042
4043  014734  004037  033164               JSR     R0,@#CLAREA     ;CLEAR 256. WORDS, FROM WRFROM
4044  014740  002370                       WRFROM                  ;STARTING FROM WRFROM
4045  014742  000400                       256.                    ;256. WORDS
4046  014744  177777                       -1                      ;FILL WITH -1
4047
4048
4049                                        ;*READ COMMAND IS LOADED
4050
4051  014746  004037  035276               JSR     R0,@#RUN        ;SETUP TO RUN FOR DATA COMMAND
4052  014752  000000                       0                       ;CYLINDER 0
4053  014754     001                       .BYTE   1.              ;SECTOR 1.
4054  014755     000                       .BYTE   0               ;TRACK 0
4055  014756  177777                       -1                      ;WORD COUNT = 1
4056  014760  003434                       REINTO                  ;BUS ADDRESS
4057                                                                 ;STARTING ADDRESS OF DATA
4058                                                                 ;BUFFER = REINTO
4059  014762  000000                       0                       ;DO NOT INHIBIT BUS ADDRESS INCREMENT
4060  014764  010000                       FMT22                   ;16 BITS PER WORD FORMAT
4061                                                                 ;DO NOT INHIBIT ECC CORRECTION
4062                                                                 ;DO NOT INHIBIT HEADER COMPARE
4063  014766  002346                       READAT                  ;GET READY TO DO A READAT
4064                                                  ;READ DATA WITH 70 IN RHCS1
4065
4066
4067                                        ;*SAVE REGISTERS FOR COMPARISON AFTER READ DATA
4068  014770  004037  033462               JSR     R0,@#SAVER      ;SAVE REGISTERS
4069  014774  002172                       RHWC                    ;RHWC IS THE FIRST REGISTER SAVED
4070  014776  004512                       SAVERE                  ;STARTING ADDRES OF WHERE
4071                                                                 ;THE REGISTERS ARE SAVED
4072  015000  000022                       18.                     ;NUMBER OF REGISTERS
4073                                                                 ;SAVED = 18.
4074
4075  015002  004737  033374               JSR     PC,@#CHECKT     ;CHECKS DVA,RDY,MOL,DPR,DRY AND VV = 1
4076                                                                 ;AND THAT ALL STATUS BITS ARE = 0
4077  015006  104401  057465               TYPE    ,CPHALT         ;CANNOT CONTINUE TESTING IF NOT
4078  015012  000000                       HALT                    ;STOP TEST
4079
4080  015014  013777  004506  165144       MOV     @#RP4VEC,@RPVEC ;SET RP04 VECTOR ADDRESS
4081                                                                 ;TO 'TIME1' IF P-CLOCK IS PRESENT
4082                                                                 ;OR TO 'TIME2' IF P-CLOCK IS NOT THERE
4083                                                                 ;'TIME' WILL ONLY SAVE
4084                                                                 ;CURRENT CYLINDER ADDRESS
4085                                                                 ;AND LOOK AHEAD REGISTERS
```

```
4086
4087
4088  015022  013746  002346                MOV     @#READAT,-(SP)  ;GET READY TO MOVE COMMAND
4089  015026  052716  000101                BIS     #GO!IE,(SP)     ;GET READY TO SET GO AND
4090                                                                  ;ENABLE INTERRUPT
4091  015032  012677  165142                MOV     (SP)+,@RHCS1    ;GO WITH
4092                                                                   ;70 IN RHCS1 FOR READ DATA
4093                                                                  ;WITH INTERRUPT ENABLED
4094
4095                                        ;*TIME IS NOT CRITICAL
4096
4097  015036  104413                        WAT                     ;WAIT FOR RDY BIT TO SET
4098  015040  002200                        RHCS1                   ;WAIT FOR RHCS1 REGISTER
4099  015042  000200                        RDY                     ;WAIT FOR RDY BIT IN RHCS1 REGISTER
4100  015044  010110                        4168.                   ;ALLOW 41680 MICRO SECONDS
4101  015046  001502                        834.                    ;RDY MUST SET BETWEEN
4102                                                                  ;33340 AND 50020 MICRO SECONDS
4103
4104                                        ;*NOW CHANGE SAVED REGISTERS TO EXPECTED VALUE
4105
4106  015050  004037  034204                JSR     R0,@#CHREG      ;CHANGE BITS IN SAVED REGISTER
4107  015054  002222                        RHDS1                   ;CHANGE RHDS1 REGISTER
4108
4109  015056  000002                        2                       ;2 BIT/BITS TO BE CHANGED
4110  015060  000001                        1                       ;NEW VALUE OF ATA IS 1
4111  015062  100000                        ATA                     ;CHANGE ATA BIT
4112  015064  000001                        1                       ;NEW VALUE OF ERR IS 1
4113  015066  040000                        ERR                     ;CHANGE ERR BIT
4114
4115  015070  004037  034204                JSR     R0,@#CHREG      ;CHANGE BITS IN SAVED REGISTER
4116  015074  002200                        RHCS1                   ;CHANGE RHCS1 REGISTER
4117
4118  015076  000002                        2                       ;2 BIT/BITS TO BE CHANGED
4119  015100  000001                        1                       ;NEW VALUE OF SC IS 1
4120  015102  100000                        SC                      ;CHANGE SC BIT
4121  015104  000001                        1                       ;NEW VALUE OF TRE IS 1
4122  015106  040000                        TRE                     ;CHANGE TRE BIT
4123
4124  015110  004037  034204                JSR     R0,@#CHREG      ;CHANGE BITS IN SAVED REGISTER
4125  015114  002202                        RHER1                   ;CHANGE RHER1 REGISTER
4126
4127  015116  000001                        1                       ;1 BIT/BITS TO BE CHANGED
4128  015120  000001                        1                       ;NEW VALUE OF HCE IS 1
4129  015122  000200                        HCE                     ;CHANGE HCE BIT
4130
4131  015124  004037  033216                JSR     R0,@#FILLRE     ;MOV 2 INTO SAVED RHDST
4132  015130  002204                        RHDST                   ;SAVED REGISTER TO CHANGE
4133  015132  000002                        2                       ;DATA
4134
4135
4136  015134  053737  004644  004536        BIS     @#ATTENT,@#SAVERE+24 ;SET APPROPIATE 'ATA' BITS
4137                                                                  ;FOR WORKING DRIVE IN
4138                                                                  ;SAVED RHAS LOCATION
4139
4140                                        ;*COMPARE ALL REGISTERS
4141
```

```
4142  015142  004037  034312              JSR     R0,@#COMREG     ;COMPARE SAVED REGISTERS WITH
4143                                                              ;PRESENT VALUE
4144  015146  004512                      SAVERE                  ;GOOD DATA SAVED IN 'SAVERE'
4145  015150  002254                      WC                      ;TEST DATA STARTING FROM 'RHWC'
4146  015152  000022                      18.                     ;18. REGISTERS TO BE COMPARED
4147  015154  015160                      3$                      ;RETURN TO 3$ ON ERROR
4148  015156  015164                      4$                      ;RETURN TO 4$ ON NO ERROR
4149
4150  015160  104047              3$:     ERROR   47              ;READING ON NON EXISTANT SECTOR
4151  015162  000207                      RTS     PC
4152                                                              ;CAUSED AN ERROR
4153                                                              ;GOOD DATA GIVES WHAT SHOULD
4154                                                              ;BE THERE
4155                                                              ;RECEIVED DATA GIVES WHAT
4156                                                              ;WAS THERE AFTER READ
4157
4158                                      ;*READ DATA WILL BE COMPARED
4159
4160  015164                      4$:
4161
4162  015164  004037  035342              JSR     R0,@#COMPAR     ;COMPARE TWO BLOCKS OF MEMORY
4163  015170  002370                      WRFROM                  ;GOOD DATA STARTS FROM WRFROM
4164  015172  003434                      REINTO                  ;TEST DATA STARTS FROM REINTO
4165  015174  000400                      256.                    ;256., WORDS TO BE COMPARED
4166  015176  015202                      5$                      ;RETURN TO 5$ ON ERROR
4167  015200  015206                      6$                      ;RETURN TO 6$ ON NO ERROR
4168
4169  015202  104050              5$:     ERROR   50              ;DATA READ FROM NON
4170  015204  000207                      RTS     PC              ;EXISTANT SECTOR CAUSED AN ERROR
4171  015206                      6$:
4172
```

```
                                ;;*****************************************************************
                                ;*TEST 15       HEADER COMPARE ERROR - RHER1 BIT #7 (HCE)

                                ;*      WRITE HEADER AND DATA IS USED TO REMOVE SECTOR 1
                                ;*      AND PUT SECTOR 0 ON CYLINDER 0
                                ;*      THEN A WRITE DATA IS GIVEN FOR SECTOR 1, TRACK 0, CYLINDER 0
                                ;*      FOR 266. WORDS
                                ;*      HCE - BIT 7 IN RHER1 SHOULD SET
                                ;*      ALL REGISTERS ARE CHECKED
                                ;*      THEN A READ HEADER AND DATA SECTOR 1, TRACK 0, CYLINDER 0
                                ;*      IS GIVEN, HCE - BIT 7 SHOULD SET AND ALL
                                ;*      HEADER AND DATA SHOULD BE READ

                                ;;*****************************************************************
015206  000004                  TST15:  SCOPE
015210  012706  001000                  MOV     #STACK,SP       ;RESET STACK
015214  012737  000015  004504          MOV     #15,@#TSTNM     ;SAVE TEST NUMBER

015222  004737  033314                  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 AND DATA

015226  004037  033140                  JSR     R0,@#FLHEAD     ;SAVE HEADER DATA IN WRFROM
015232  002370                          WRFROM                  ;LOCATION WHERE SAVED
015234  000006                          6                       ;NUMBER OF WORDS SAVED
015236  010000                          10000                   ;FIRST DATA WORD
015240  000000                          0                       ;SECOND DATA WORD
015242  000000                          0                       ;THIRD DATA WORD
015244  000000                          0                       ;FOURTH DATA WORD
015246  000001                          1                       ;FIFTH DATA WORD
015250  000001                          1                       ;SIXTH DATA WORD

                                        ;*FILL READ INTO BUFFER WITH ALL ONES

015252  004037  033164                  JSR     R0,@#CLAREA     ;CLEAR 256. WORDS, FROM REINTO
015256  003434                          REINTO                  ;STARTING FROM REINTO
015260  000400                          256.                    ;256. WORDS
015262  177777                          -1                      ;FILL WITH -1

                                        ;*WRITE HEADER AND DATA IS LOADED

015264  004037  035276                  JSR     R0,@#RUN        ;SETUP TO RUN FOR DATA COMMAND
015270  000000                          0                       ;CYLINDER 0
015272     001                          .BYTE   1               ;SECTOR 1
015273     000                          .BYTE   0               ;TRACK 0
015274  177772                          -2-4                    ;WORD COUNT (DATA) = 2 +
                                                                ;4 HEADER WORDS
015276  002370                          WRFROM                  ;BUS ADDRESS
                                                                ;STARTING ADDRESS OF DATA
                                                                ;BUFFER = WRFROM
015300  000000                          0                       ;DO NOT INHIBIT BUS ADDRESS INCREMENT
015302  010000                          FMT22                   ;16 BITS PER WORD FORMAT
```

```
                                                        ;DO NOT INHIBIT ECC CORRECTION
                                                        ;DO NOT INHIBIT HEADER COMPARE
015304  002344                          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

015306  004037  033462                  JSR     R0,@#SAVER      ;SAVE REGISTERS
015312  002172                          RHWC                    ;RHWC IS THE FIRST REGISTER SAVED
015314  004512                          SAVERE                  ;STARTING ADDRES OF WHERE
                                                                ;THE REGISTERS ARE SAVED
015316  000021                          17.                     ;NUMBER OF REGISTERS
                                                                ;SAVED = 17.

015320  004737  033374                  JSR     PC,@#CHECKT     ;CHECKS DVA,RDY,MOL,DPR,DRY AND VV = 1
                                                                ;AND THAT ALL STATUS BITS ARE = 0
015324  104401  057465                  TYPE    ,CPHALT         ;CANNOT CONTINUE TESTING IF NOT
015330  000000                          HALT                    ;STOP TEST

015332  013777  004506  164626          MOV     @#RP4VEC,@RPVEC ;SET RP04 VECTOR ADDRESS
                                                                ;TO 'TIME1' IF P-CLOCK IS PRESENT
                                                                ;OR TO 'TIME2' IF P-CLOCK IS NOT THERE
                                                                ;'TIME' WILL ONLY SAVE
                                                                ;CURRENT CYLINDER ADDRESS
                                                                ;AND LOOK AHEAD REGISTERS


015340  013746  002344                  MOV     @#WRIFOR,-(SP)  ;GET READY TO MOVE COMMAND
015344  052716  000101                  BIS     #GO!IE,(SP)     ;GET READY TO SET GO AND
                                                                ;ENABLE INTERRUPT
015350  012677  164624                  MOV     (SP)+,@RHCS1    ;GO WITH
                                                                 ;62 IN RHCS1 FOR WRITE HEADER AND DATA
                                                                ;WITH INTERRUPT ENABLED


                                        ;*TIME IS NOT CRITICAL

015354  104413                          WAT                     ;WAIT FOR RDY BIT TO SET
015356  002200                          RHCS1                   ;WAIT FOR RHCS1 REGISTER
015360  000200                          RDY                     ;WAIT FOR RDY BIT IN RHCS1 REGISTER
015362  004704                          2500.                   ;ALLOW 25000 MICRO SECONDS
015364  004704                          2500.                   ;RDY MUST SET BETWEEN
                                                                ;00 AND 50000 MICRO SECONDS


                                        ;*NOW CHANGE SAVED REGISTERS TO EXPECTED VALUE

015366  004037  033216                  JSR     R0,@#FILLRE     ;MOV 0 INTO SAVED RHWC
015372  002172                          RHWC                    ;SAVED REGISTER TO CHANGE
015374  000000                          0                       ;DATA


015376  004037  033216                  JSR     R0,@#FILLRE     ;MOV WRFROM+<6*2> INTO SAVED RHBA
015402  002174                          RHBA                    ;SAVED REGISTER TO CHANGE
015404  002404                          WRFROM+<6*2>                    ;DATA
```

```
4285
4286  015406  004037  033216              JSR     R0,@#FILLRE     ;MOV 2 INTO SAVED RHDST
4287  015412  002204                      RHDST                   ;SAVED REGISTER TO CHANGE
4288  015414  000002                      2                       ;DATA
4289
4290
4291                                      ;*COMPARE ALL REGISTERS
4292
4293  015416  004037  034312              JSR     R0,@#COMREG     ;COMPARE SAVED REGISTERS WITH
4294                                                              ;PRESENT VALUE
4295  015422  004512                      SAVERE                  ;GOOD DATA SAVED IN 'SAVERE'
4296  015424  002254                      WC                      ;TEST DATA STARTING FROM 'RHWC'
4297  015426  000021                      17.                     ;17. REGISTERS TO BE COMPARED
4298  015430  015434                      1$                      ;RETURN TO 1$ ON ERROR
4299  015432  015440                      2$                      ;RETURN TO 2$ ON NO ERROR
4300
4301  015434  104027              1$:     ERROR   27              ;WRITING HEADER AND DATA CAUSED
4302  015436  000207                      RTS     PC              ;IMPROPER REGISTER CHANGE
4303                                                              ;GOOD DATA GIVES WHAT
4304                                                              ;SHOULD BE THERE
4305                                                              ;RECEIVED DATA GIVES WHAT
4306                                                              ;WAS THERE AFTER WRITE
4307
```

```
                                                ;*NOW A WRITE DATA WILL BE DONE ON SAME CYLINDER, SECTOR
                                                ;*TRACK
                                
015440                                  2$:
                                                ;*FILL WRITE FROM BUFFER WITH DATA
                                
015440  004037  033164                        JSR     R0,@#CLAREA     ;CLEAR 266. WORDS, FROM WRFROM
015444  002370                                WRFROM                  ;STARTING FROM WRFROM
015446  000412                                266.                    ;266. WORDS
015450  177400                                177400                  ;FILL WITH 177400
                                
                                                ;*WRITE DATA COMMAND IS LOADED
                                
015452  004037  035276                        JSR     R0,@#RUN        ;SETUP TO RUN FOR DATA COMMAND
015456  000000                                0                       ;CYLINDER 0
015460     001                                .BYTE   1               ;SECTOR 1
015461     000                                .BYTE   0               ;TRACK 0
015462  177366                                -266.                   ;WORD COUNT = 266.
015464  002370                                WRFROM                  ;BUS ADDRESS
                                                                        ;STARTING ADDRESS OF DATA
                                                                        ;BUFFER = WRFROM
015466  000000                                0                       ;DO NOT INHIBIT BUS ADDRESS INCREMENT
015470  010000                                FMT22                   ;16 BITS PER WORD FORMAT
                                                                        ;DO NOT INHIBIT ECC CORRECTION
                                                                        ;DO NOT INHIBIT HEADER COMPARE
015472  002342                                WRIDAT                  ;GET READY TO DO A WRIDAT
                                                                ;WRITE DATA WITH 60 IN RHCS1
                                
                                
                                                ;*SAVE REGISTERS FOR COMPARISON AFTER WRITE DATA
015474  004037  033462                        JSR     R0,@#SAVER      ;SAVE REGISTERS
015500  002172                                RHWC                    ;RHWC IS THE FIRST REGISTER SAVED
015502  004512                                SAVERE                  ;STARTING ADDRES OF WHERE
                                                                        ;THE REGISTERS ARE SAVED
015504  000022                                18.                     ;NUMBER OF REGISTERS
                                                                        ;SAVED = 18.
                                
015506  004737  033374                        JSR     PC,@#CHECKT     ;CHECKS DVA,RDY,MOL,DPR,DRY AND VV = 1
                                                                        ;AND THAT ALL STATUS BITS ARE = 0
015512  104401  057465                        TYPE    ,CPHALT         ;CANNOT CONTINUE TESTING IF NOT
015516  000000                                HALT                    ;STOP TEST
                                
015520  013777  004506  164440                MOV     @#RP4VEC,@RPVEC ;SET RP04 VECTOR ADDRESS
                                                                        ;TO 'TIME1' IF P-CLOCK IS PRESENT
                                                                        ;OR TO 'TIME2' IF P-CLOCK IS NOT THERE
                                                                        ;'TIME' WILL ONLY SAVE
                                                                        ;CURRENT CYLINDER ADDRESS
                                                                        ;AND LOOK AHEAD REGISTERS
                                
                                
015526  013746  002342                        MOV     @#WRIDAT,-(SP)  ;GET READY TO MOVE COMMAND
015532  052716  000101                        BIS     #GO!IE,(SP)     ;GET READY TO SET GO AND
                                                                        ;ENABLE INTERRUPT
                                
015536  012677  164436                        MOV     (SP)+,@RHCS1    ;GO WITH
```

```
                                                        ;60 IN RHCS1 FOR WRITE DATA
                                                       ;WITH INTERRUPT ENABLED

                                    ;*TIME IS NOT CRITICAL

015542  104413                      WAT                     ;WAIT FOR RDY BIT TO SET
015544  002200                      RHCS1                   ;WAIT FOR RHCS1 REGISTER
015546  000200                      RDY                     ;WAIT FOR RDY BIT IN RHCS1 REGISTER
015550  001614                      908.                    ;ALLOW 9080 MICRO SECONDS
015552  001502                      834.                    ;RDY MUST SET BETWEEN
                                                            ;740 AND 17420 MICRO SECONDS

                                    ;*NOW CHANGE REGISTERS TO EXPECTED VALUE
015554  005737  004640              TST     @#RH70          ;RUNNING ON RH70 ?
015560  001421                      BEQ     JP1             ;IF NOT, SKIP NEXT RH70 CODE

015562  004037  033216              JSR     R0,@#FILLRE     ;MOV -258. INTO SAVED RHWC
015566  002172                      RHWC                    ;SAVED REGISTER TO CHANGE
015570  177376              VAR1:   -258.                   ;DATA

015572  004037  033216              JSR     R0,@#FILLRE     ;MOV WRFROM+<8.*2> INTO SAVED AREA
015576  002174                      RHBA                    ;SAVED REGISTER TO CHANGE
015600  002410              VAR2:   WRFROM+<8.*2>                   ;DATA

015602  004037  034204              JSR     R0,@#CHREG      ;CHANGE BITS IN SAVED REGISTER
015606  002176                      RHCS2                   ;CHANGE RHCS2 REGISTER

015610  000002                      2                       ;2 BIT/BITS TO BE CHANGED
015612  000001                      1                       ;NEW VALUE OF OR IS 1
015614  000200                      OR                      ;CHANGE OR BIT
015616  000000                      0                       ;NEW VALUE OF IR IS 0
015620  000100                      IR                      ;CHANGE IR BIT
015622  000416              VAR5:   BR      JP2             ;SKIP NEXT RH11 CODE

015624                      JP1:

015624  004037  033216              JSR     R0,@#FILLRE     ;MOV -200. INTO SAVED RHWC
015630  002172                      RHWC                    ;SAVED REGISTER TO CHANGE
015632  177470                      -200.                   ;DATA

015634  004037  033216              JSR     R0,@#FILLRE     ;MOV WRFROM+<66.*2> INTO SAVED RHBA
015640  002174                      RHBA                    ;SAVED REGISTER TO CHANGE
015642  002574                      WRFROM+<66.*2>                  ;DATA

015644  004037  034204              JSR     R0,@#CHREG      ;CHANGE BITS IN SAVED REGISTER
015650  002176                      RHCS2                   ;CHANGE RHCS2 REGISTER

015652  000001                      1                       ;1 BIT/BITS TO BE CHANGED
015654  000001                      1                       ;NEW VALUE OF OR IS 1
015656  000200                      OR                      ;CHANGE OR BIT

015660                      JP2:
```

```
 4420  015660  004037  034204                JSR     R0,@#CHREG      ;CHANGE BITS IN SAVED REGISTER
 4421  015664  002200                        RHCS1                   ;CHANGE RHCS1 REGISTER
 4422
 4423  015666  000002                        2                       ;2 BIT/BITS TO BE CHANGED
 4424  015670  000001                        1                       ;NEW VALUE OF SC IS 1
 4425  015672  100000                        SC                      ;CHANGE SC BIT
 4426  015674  000001                        1                       ;NEW VALUE OF TRE IS 1
 4427  015676  040000                        TRE                     ;CHANGE TRE BIT
 4428
 4429  015700  004037  033216                JSR     R0,@#FILLRE     ;MOV 2 INTO SAVED RHDST
 4430  015704  002204                        RHDST                   ;SAVED REGISTER TO CHANGE
 4431  015706  000002                        2                       ;DATA
 4432
 4433
 4434  015710  053737  004644  004536        BIS     @#ATTENT,@#SAVERE+24 ;SET APPROPIATE 'ATA' BITS
 4435                                                                ;FOR WORKING DRIVE IN
 4436                                                                ;SAVED RHAS LOCATION
 4437
 4438  015716  004037  034204                JSR     R0,@#CHREG      ;CHANGE BITS IN SAVED REGISTER
 4439  015722  002202                        RHER1                   ;CHANGE RHER1 REGISTER
 4440
 4441  015724  000001                        1                       ;1 BIT/BITS TO BE CHANGED
 4442  015726  000001                        1                       ;NEW VALUE OF HCE IS 1
 4443  015730  000200                        HCE                     ;CHANGE HCE BIT
 4444
 4445  015732  004037  034204                JSR     R0,@#CHREG      ;CHANGE BITS IN SAVED REGISTER
 4446  015736  002222                        RHDS1                   ;CHANGE RHDS1 REGISTER
 4447
 4448  015740  000002                        2                       ;2 BIT/BITS TO BE CHANGED
 4449  015742  000001                        1                       ;NEW VALUE OF ATA IS 1
 4450  015744  100000                        ATA                     ;CHANGE ATA BIT
 4451  015746  000001                        1                       ;NEW VALUE OF ERR IS 1
 4452  015750  040000                        ERR                     ;CHANGE ERR BIT
 4453
 4454                                        ;*COMPARE ALL REGISTERS
 4455
 4456  015752  004037  034312                JSR     R0,@#COMREG     ;COMPARE SAVED REGISTERS WITH
 4457                                                                ;PRESENT VALUE
 4458  015756  004512                        SAVERE                  ;GOOD DATA SAVED IN 'SAVERE'
 4459  015760  002254                        WC                      ;TEST DATA STARTING FROM 'RHWC'
 4460  015762  000022                        18.                     ;18. REGISTERS TO BE COMPARED
 4461  015764  015770                        3$                      ;RETURN TO 3$ ON ERROR
 4462  015766  015774                        4$                      ;RETURN TO 4$ ON NO ERROR
 4463
 4464  015770  104052              3$:       ERROR   52              ;WRITE DATA ON NON EXISTANT SECTOR
 4465  015772  000207                        RTS     PC              ;CAUSED IMPROPER REGISTER CHANGE
 4466                                                                ;ATTEMPTED WRITE WAS ON
 4467                                                                ;CYLINDER 0,SECTOR 1, TRACK 0
 4468                                                                ;GOOD DATA GIVES WHAT SHOULD BE THERE
 4469                                                                ;RECEIVED DATA GIVES WHAT WAS THERE
 4470                                                                ;AFTER COMMAND
 4471
 4472
 4473                                        ;*READ HEADER AND DATA SECTOR 1, TRACK 0, CYLINDER 0
 4474                                        ;*WILL BE ATTEMPTED
 4475  015774                        4$:
```

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

016000  004037  033140                  JSR     R0,@#FLHEAD     ;SAVE HEADER DATA IN WRFROM
016004  002370                          WRFROM                  ;LOCATION WHERE SAVED
016006  000006                          6                       ;NUMBER OF WORDS SAVED
016010  010000                          10000                   ;FIRST DATA WORD
016012  000000                          0                       ;SECOND DATA WORD
016014  000000                          0                       ;THIRD DATA WORD
016016  000000                          0                       ;FOURTH DATA WORD
016020  000001                          1                       ;FIFTH DATA WORD
016022  000001                          1                       ;SIXTH DATA WORD

016024  004037  033164                  JSR     R0,@#CLAREA     ;CLEAR 198. WORDS, FROM WRFROM+<6*2>
016030  002404                          WRFROM+<6*2>                    ;STARTING FROM WRFROM+<6*2>
016032  000306                          198.                    ;198. WORDS
016034  000000                          0                       ;FILL WITH 0

016036  004037  033164                  JSR     R0,@#CLAREA     ;CLEAR 62. WORDS, FROM WRFROM+<204.*2>
016042  003220                          WRFROM+<204.*2>                 ;STARTING FROM WRFROM+<204.*2>
016044  000076                          62.                     ;62. WORDS
016046  177777                          -1                      ;FILL WITH -1


                                        ;*FILL READ INTO BUFFER WITH ALL ONES

016050  004037  033164                  JSR     R0,@#CLAREA     ;CLEAR 256. WORDS, FROM REINTO
016054  003434                          REINTO                  ;STARTING FROM REINTO
016056  000400                          256.                    ;256. WORDS
016060  177777                          -1                      ;FILL WITH -1

                                        ;*FILL REGISTERS WITH READ HEADER AND DATA COMMAND

016062  004037  035276                  JSR     R0,@#RUN        ;SETUP TO RUN FOR DATA COMMAND
016066  000000                          0                       ;CYLINDER 0
016070     001                          .BYTE   1               ;SECTOR 1
016071     000                          .BYTE   0               ;TRACK 0
016072  177464                          -200.-4                 ;WORD COUNT (DATA) = 200. +
                                                                ;4 HEADER WORDS
016074  003434                          REINTO                  ;BUS ADDRESS
                                        ;                       ;STARTING ADDRESS OF DATA
                                                                ;BUFFER = REINTO
016076  000000                          0                       ;DO NOT INHIBIT BUS ADDRESS INCREMENT
016100  014000                          ECI!FMT22                       ;16 BITS PER WORD FORMAT
                                                                ;INHIBIT ECC CORRECTION
                                                                ;DO NOT INHIBIT HEADER COMPARE
016102  002350                          REFOR                   ;GET READY TO DO A REFOR
                                                                ;READ HEADER AND DATA WITH 72 IN RHCS1
```

```
                                        ;*SAVE REGISTERS FOR COMPARISON AFTER READ
                                        ;*HEADER AND DATA
016104  004037  033462                  JSR     R0,@#SAVER      ;SAVE REGISTERS
016110  002172                          RHWC                    ;RHWC IS THE FIRST REGISTER SAVED
016112  004512                          SAVERE                  ;STARTING ADDRES OF WHERE
                                                                ;THE REGISTERS ARE SAVED
016114  000022                          18.                     ;NUMBER OF REGISTERS
                                                                ;SAVED = 18.

016116  004737  033374                  JSR     PC,@#CHECKT     ;CHECKS DVA,RDY,MOL,DPR,DRY AND VV = 1
                                                                ;AND THAT ALL STATUS BITS ARE = 0
016122  104401  057465                  TYPE    ,CPHALT         ;CANNOT CONTINUE TESTING IF NOT
016126  000000                          HALT                    ;STOP TEST

016130  013777  004506  164030          MOV     @#RP4VEC,@RPVEC ;SET RP04 VECTOR ADDRESS
                                                                ;TO 'TIME1' IF P-CLOCK IS PRESENT
                                                                ;OR TO 'TIME2' IF P-CLOCK IS NOT THERE
                                                                ;'TIME' WILL ONLY SAVE
                                                                ;CURRENT CYLINDER ADDRESS
                                                                ;AND LOOK AHEAD REGISTERS

016136  013746  002350                  MOV     @#REFOR,-(SP)   ;GET READY TO MOVE COMMAND
016142  052716  000101                  BIS     #GO!IE,(SP)     ;GET READY TO SET GO AND
                                                                ;ENABLE INTERRUPT
016146  012677  164026                  MOV     (SP)+,@RHCS1    ;GO WITH
                                                                  ;72 IN RHCS1 FOR READ DATA
                                                                ;WITH INTERRUPT ENABLED

                                        ;*TIME IS NOT CRITICAL

016152  104413                          WAT                     ;WAIT FOR RDY BIT TO SET
016154  002200                          RHCS1                   ;WAIT FOR RHCS1 REGISTER
016156  000200                          RDY                     ;WAIT FOR RDY BIT IN RHCS1 REGISTER
016160  001614                          908.                    ;ALLOW 9080 MICRO SECONDS
016162  001507                          839.                    ;RDY MUST SET BETWEEN
                                                                ;690 AND 17470 MICRO SECONDS

                                        ;*CHANGE SAVED REGISTERS TO EXPECTED VALUE

016164  004037  033216                  JSR     R0,@#FILLRE     ;MOV 0 INTO SAVED RHWC
016170  002172                          RHWC                    ;SAVED REGISTER TO CHANGE
016172  000000                          0                       ;DATA

016174  004037  033216                  JSR     R0,@#FILLRE     ;MOV REINTO+<204.*2> INTO SAVED RHBA
016200  002174                          RHBA                    ;SAVED REGISTER TO CHANGE
016202  004264                          REINTO+<204.*2>                 ;DATA

016204  004037  034204                  JSR     R0,@#CHREG      ;CHANGE BITS IN SAVED REGISTER
016210  002202                          RHER1                   ;CHANGE RHER1 REGISTER

016212  000001                          1                       ;1 BIT/BITS TO BE CHANGED
016214  000001                          1                       ;NEW VALUE OF HCE IS 1
016216  000200                          HCE                     ;CHANGE HCE BIT
```

```
4588
4589  016220  004037  034204          JSR     R0,@#CHREG      ;CHANGE BITS IN SAVED REGISTER
4590  016224  002222                  RHDS1                   ;CHANGE RHDS1 REGISTER
4591
4592  016226  000002                  2                       ;2 BIT/BITS TO BE CHANGED
4593  016230  000001                  1                       ;NEW VALUE OF ATA IS 1
4594  016232  100000                  ATA                     ;CHANGE ATA BIT
4595  016234  000001                  1                       ;NEW VALUE OF ERR IS 1
4596  016236  040000                  ERR                     ;CHANGE ERR BIT
4597
4598  016240  004037  033216          JSR     R0,@#FILLRE     ;MOV 2 INTO SAVED RHDST
4599  016244  002204                  RHDST                   ;SAVED REGISTER TO CHANGE
4600  016246  000002                  2                       ;DATA
4601
4602
4603  016250  004037  034204          JSR     R0,@#CHREG      ;CHANGE BITS IN SAVED REGISTER
4604  016254  002200                  RHCS1                   ;CHANGE RHCS1 REGISTER
4605
4606  016256  000001                  1                       ;1 BIT/BITS TO BE CHANGED
4607  016260  000001                  1                       ;NEW VALUE OF SC!TRE IS 1
4608  016262  140000                  SC!TRE                  ;CHANGE SC!TRE BIT
4609
4610  016264  053737  004644  004536  BIS     @#ATTENT,@#SAVERE+24 ;SET APPROPIATE 'ATA' BITS
4611                                                          ;FOR WORKING DRIVE IN
4612                                                          ;SAVED RHAS LOCATION
4613
4614                                  ;*COMPARE REGISTERS BEFORE READ HEADER AND DATA
4615                                  ;*WITH AFTER
4616
4617  016272  004037  034312          JSR     R0,@#COMREG     ;COMPARE SAVED REGISTERS WITH
4618                                                          ;PRESENT VALUE
4619  016276  004512                  SAVERE                  ;GOOD DATA SAVED IN 'SAVERE'
4620  016300  002254                  WC                      ;TEST DATA STARTING FROM 'RHWC'
4621  016302  000022                  18.                     ;18. REGISTERS TO BE COMPARED
4622  016304  016310                  5$                      ;RETURN TO 5$ ON ERROR
4623  016306  016314                  6$                      ;RETURN TO 6$ ON NO ERROR
4624
4625
4626  016310  104031          5$:     ERROR   31              ;READ HEADER AND DATA WITH
4627  016312  000207                  RTS     PC              ;FORCED HEADER COMPARE ERROR
4628                                                          ;CAUSED ERROR
4629                                                          ;GOOD DATA GIVES WHAT SHOULD
4630                                                          ;BE THERE
4631                                                          ;RECEIVED DATA GIVES WHAT
4632                                                          ;WAS THERE AFTER READ
4633
4634                                  ;*NOW COMPARE READ DATA
4635                                  ;*THE COMMAND READ ONLY 204 WORDS, 4 HEADER WORDS
4636                                  ;*AND 200 DATA WORDS
4637
4638  016314                  6$:
4639
4640  016314  004037  035342          JSR     R0,@#COMPAR     ;COMPARE TWO BLOCKS OF MEMORY
4641  016320  002370                  WRFROM                  ;GOOD DATA STARTS FROM WRFROM
4642  016322  003434                  REINTO                  ;TEST DATA STARTS FROM REINTO
4643  016324  000400                  256.                    ;256., WORDS TO BE COMPARED
```

```
4644  016326  016332                    7$                  ;RETURN TO 7$ ON ERROR
4645  016330  016336                    8$                  ;RETURN TO 8$ ON NO ERROR
4646
4647  016332  104034              7$:   ERROR   34          ;DATA READ FROM A FORCED
4648  016334  000207                    RTS     PC          ;HEADER COMPARE ERROR IS
4649                                                        ;INCORRECT
4650                                                        ;GOOD DATA GIVES WHAT
4651                                                        ;THE READ HEADER AND DATA
4652                                                        ;SHOULD HAVE READ
4653                                                        ;BAD DATA GIVES WHAT
4654                                                        ;WAS IN BUFFER AFTER
4655                                                        ;READ COMMAND
4656  016336                    8$:
4657
4658
4659
4660
4661
4662
```

```
                                ;;*****************************************************************
                                ;*TEST 16       HEADER COMPARE ERROR - RHER1 HCE
                                ;*
                                ;*      WITH THE HEADS ON CYLINDER 0 A SEARCH COMMAND IS GIVEN
                                ;*      FOR CYLINDER 0 TRACK 0 SECTOR 1, ALTHOUGH THE HEADER
                                ;*      FOR THIS SECTOR IS CHANGED TO SECTOR 0 HCE-BIT #7
                                ;*      IN RHER1 SHOULD NOT SET
                                ;*      BECAUSE SEARCH DOES NOT READ HEADER BUT ONLY USES SECTOR COUNTER
                                ;;*****************************************************************
016336  000004                  TST16:  SCOPE
016340  012706  001000                  MOV     #STACK,SP       ;RESET STACK
016344  012737  000016  004504          MOV     #16,@#TSTNM     ;SAVE TEST NUMBER
016352  004737  033314                  JSR     PC,@#CLDISK     ;SET R1-RHCS1, R2-RHCS2
                                                                ;R3-RHDS1, R4-RHER1
                                                                ;GIVE RH-11 INITIALIZE
                                                                ;SETUP UNIT NUMBER
                                        ;*GET HEADS TO CYLINDER 0
016356  004737  033374                  JSR     PC,@#CHECKT     ;CHECKS DVA,RDY,MOL,DPR,DRY AND VV = 1
                                                                ;AND THAT ALL STATUS BITS ARE = 0
016362  104401  057465                  TYPE    ,CPHALT         ;CANNOT CONTINUE TESTING IF NOT
016366  000000                          HALT                    ;STOP TEST
016370  013777  004506  163570          MOV     @#RP4VEC,@RPVEC ;SET RP04 VECTOR ADDRESS
                                                                ;TO 'TIME1' IF P-CLOCK IS PRESENT
                                                                ;OR TO 'TIME2' IF P-CLOCK IS NOT THERE
                                                                ;'TIME' WILL ONLY SAVE
                                                                ;CURRENT CYLINDER ADDRESS
                                                                ;AND LOOK AHEAD REGISTERS


016376  013746  002326          MOV     @#RECAL1,-(SP)  ;GET READY TO MOVE COMMAND
016402  052716  000101          BIS     #GO!IE,(SP)     ;GET READY TO SET GO AND
                                                        ;ENABLE INTERRUPT
016406  012677  163566          MOV     (SP)+,@RHCS1    ;GO WITH
                                                         ;6 IN RHCS1 FOR RECALIBRATE
                                                        ;WITH INTERRUPT ENABLED
016412  011100                  MOV     @R1,R0          ;SAVE RHCS1 DURING ABOVE OPERATION
016414  011305                  MOV     @R3,R5          ;SAVE RHDS1 DURING ABOVE OPERATION


016416  104413                  WAT                     ;WAIT FOR DRY BIT TO SET
016420  002222                  RHDS1                   ;WAIT FOR RHDS1 REGISTER
016422  000200                  DRY                     ;WAIT FOR DRY BIT IN RHDS1 REGISTER
016424  012740                  5600.                   ;ALLOW 56000 MICRO SECONDS
016426  012737                  5599.                   ;DRY MUST SET BETWEEN
                                                        ;10 AND 111990 MICRO SECONDS


016430  004737  033314          JSR     PC,@#CLDISK     ;SET R1-RHCS1, R2-RHCS2
                                                        ;R3-RHDS1, R4-RHER1
                                                        ;GIVE RH-11 INITIALIZE
                                                        ;SETUP UNIT NUMBER
```

```
4719
4720                                        ;*FILL REGISTERS FOR SEARCH
4721
4722  016434  004037  033244                JSR     R0,@#SRCH       ;SEARCH FOR
4723  016440  000000                        0                       ;CYLINDER 0
4724  016442     001              .BYTE     1                       ;SECTOR 1
4725  016443     000              .BYTE     0                       ;TRACK 0
4726
4727
4728                                        ;*SAVE REGISTERS FOR COMPARISON AFTER SEARCH
4729  016444  004037  033462                JSR     R0,@#SAVER      ;SAVE REGISTERS
4730  016450  002172                        RHWC                    ;RHWC IS THE FIRST REGISTER SAVED
4731  016452  004512                        SAVERE                  ;STARTING ADDRES OF WHERE
4732                                                                ;THE REGISTERS ARE SAVED
4733  016454  000022                        18.                     ;NUMBER OF REGISTERS
4734                                                                ;SAVED = 18.
4735
4736
4737  016456  004737  033374                JSR     PC,@#CHECKT     ;CHECKS DVA,RDY,MOL,DPR,DRY AND VV = 1
4738                                                                ;AND THAT ALL STATUS BITS ARE = 0
4739  016462  104401  057465                TYPE    ,CPHALT         ;CANNOT CONTINUE TESTING IF NOT
4740  016466  000000                        HALT                    ;STOP TEST
4741
4742  016470  013777  004506  163470        MOV     @#RP4VEC,@RPVEC ;SET RP04 VECTOR ADDRESS
4743                                                                ;TO 'TIME1' IF P-CLOCK IS PRESENT
4744                                                                ;OR TO 'TIME2' IF P-CLOCK IS NOT THERE
4745                                                                ;'TIME' WILL ONLY SAVE
4746                                                                ;CURRENT CYLINDER ADDRESS
4747                                                                ;AND LOOK AHEAD REGISTERS
4748
4749
4750  016476  013746  002334                MOV     @#SERCH,-(SP)   ;GET READY TO MOVE COMMAND
4751  016502  052716  000101                BIS     #GO!IE,(SP)     ;GET READY TO SET GO AND
4752                                                                ;ENABLE INTERRUPT
4753  016506  012677  163466                MOV     (SP)+,@RHCS1    ;GO WITH
4754                                                                ;WITH INTERRUPT ENABLED
4755  016512  011100                        MOV     @R1,R0          ;SAVE RHCS1 DURING ABOVE OPERATION
4756  016514  011305                        MOV     @R3,R5          ;SAVE RHDS1 DURING ABOVE OPERATION
4757
4758
4759  016516  104413                        WAT                     ;WAIT FOR DRY BIT TO SET
4760  016520  002222                        RHDS1                   ;WAIT FOR RHDS1 REGISTER
4761  016522  000200                        DRY                     ;WAIT FOR DRY BIT IN RHDS1 REGISTER
4762  016524  001614                        908.                    ;ALLOW 9080 MICRO SECONDS
4763  016526  001507                        839.                    ;DRY MUST SET BETWEEN
4764                                                                ;690 AND 17470 MICRO SECONDS
4765
4766                                        ;*COMPARE CONTENTS OF RHCS1 AND RHDS1 ALREADY SAVED IN
4767                                        ;*R0 AND R5 IMMEDIATELY AFTER GO IS ISSUED
4768
4769  016530  013746  002334                MOV     @#SERCH,-(SP)   ;SAVE COMMAND
4770  016534  052716  004301                BIS     #IE!GO!DVA!RDY,(SP)     ;INCLUDE IE!GO!DVA!RDY
4771  016540  011637  001124                MOV     (SP),@#$GDDAT   ;SAVE FOR PRINTOUT
4772  016544  022600                        CMP     (SP)+,R0        ;DURING ABOVE OPERATION ONLY IE!GO!DVA!RDY
4773                                                                ;AND COMMAND SHOULD BE SET
4774  016546  001405                        BEQ     67$             ;BRANCH IF GOOD
```

```
 4775  016550  010037  001126              MOV     R0,@#$BDDAT     ;BAD DATA
 4776  016554  010137  004500              MOV     R1,@#REGADR     ;FAILING REGISTER RHCS1
 4777  016560  104021                      ERROR   21              ;DURING ABOVE OPERATION ONLY
 4778                                                              ;COMMAND AND IE!GO!DVA!RDY SHOULD BE SET
 4779  016562  012746  010500      67$:    MOV     #MOL!DPR!VV,-(SP)       ;SAVE BITS SET DURING OPERATION IN RHDS1
 4780  016566  011637  001124              MOV     (SP),@#$GDDAT   ;SAVE FOR PRINTOUT
 4781  016572  022605                      CMP     (SP)+,R5        ;DURING ABOVE OPERATION ONLY MOL!DPR!VV
 4782                                                              ;SHOULD BE SET
 4783  016574  001405                      BEQ     69$             ;BRANCH IF GOOD
 4784  016576  010537  001126              MOV     R5,@#$BDDAT     ;BAD DATA
 4785  016602  010337  004500              MOV     R3,@#REGADR     ;FAILING REGISTER RHDS1
 4786  016606  104063                      ERROR   63              ;DURING ABOVE OPERATION ONLY
 4787                                                              ;MOL!DPR!VV SHOULD BE SET
 4788  016610                      69$:
 4789
 4790                                      ;*CHANGE SAVED REGISTERS TO EXPECTED VALUE
 4791
 4792  016610  004037  034204              JSR     R0,@#CHREG      ;CHANGE BITS IN SAVED REGISTER
 4793  016614  002200                      RHCS1                   ;CHANGE RHCS1 REGISTER
 4794
 4795  016616  000001                      1                       ;1 BIT/BITS TO BE CHANGED
 4796  016620  000001                      1                       ;NEW VALUE OF SC IS 1
 4797  016622  100000                      SC                      ;CHANGE SC BIT
 4798
 4799  016624  004037  034204              JSR     R0,@#CHREG      ;CHANGE BITS IN SAVED REGISTER
 4800  016630  002222                      RHDS1                   ;CHANGE RHDS1 REGISTER
 4801
 4802  016632  000001                      1                       ;1 BIT/BITS TO BE CHANGED
 4803  016634  000001                      1                       ;NEW VALUE OF ATA IS 1
 4804  016636  100000                      ATA                     ;CHANGE ATA BIT
 4805
 4806  016640  053737  004644  004536      BIS     @#ATTENT,@#SAVERE+24 ;SET APPROPIATE 'ATA' BITS
 4807                                                              ;FOR WORKING DRIVE IN
 4808                                                              ;SAVED RHAS LOCATION
 4809
 4810                                      ;*COMPARE REGISTERS BEFORE SEARCH WITH AFTER SEARCH
 4811
 4812  016646  004037  034312              JSR     R0,@#COMREG     ;COMPARE SAVED REGISTERS WITH
 4813                                                              ;PRESENT VALUE
 4814  016652  004512                      SAVERE                  ;GOOD DATA SAVED IN 'SAVERE'
 4815  016654  002254                      WC                      ;TEST DATA STARTING FROM 'RHWC'
 4816  016656  000022                      18.                     ;18. REGISTERS TO BE COMPARED
 4817  016660  016664                      1$                      ;RETURN TO 1$ ON ERROR
 4818  016662  016670                      2$                      ;RETURN TO 2$ ON NO ERROR
 4819
 4820
 4821  016664  104047              1$:     ERROR   47              ;SEARCH TO A NON-EXISTANT
 4822
 4823  016666  000207                      RTS     PC              ;SECTOR CAUSED IMPROPER
 4824                                                              ;REGISTER CHANGE
 4825                                                              ;GOOD DATA GIVES WHAT SHOULD
 4826                                                              ;BE THERE
 4827                                                              ;RECEIVED DATA GIVES
 4828                                                              ;WHAT WAS THERE AFTER
 4829                                                              ;SEARCH
 4830  016670                      2$:
```

```
 4831
```

```
                                ;;*****************************************************************
                                ;*TEST 17       RESTORE SECTOR 1 CYLINDER 1 TRACK 1

                                ;*      THIS REPLACES REMOVED SECTOR
                                ;*
                                ;*
                                ;*
                                ;*      WRITE HEADER AND DATA CYLINDER 0, FORMAT 16 BITS PER WORD
                                ;*      TRACK 0, SECTOR 1, 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
                                ;*      10000,1,0,0, 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.

                                ;;*****************************************************************
016670  000004                  TST17:  SCOPE
016672  012706  001000                  MOV     #STACK,SP       ;RESET STACK
016676  012737  000017  004504          MOV     #17,@#TSTNM     ;SAVE TEST NUMBER

016704  004737  033314                  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

016710  004037  033140                  JSR     R0,@#FLHEAD     ;SAVE HEADER DATA IN WRFROM
016714  002370                          WRFROM                  ;LOCATION WHERE SAVED
016716  000004                          4                       ;NUMBER OF WORDS SAVED
016720  010000                          10000                   ;FIRST DATA WORD
016722  000001                          1                       ;SECOND DATA WORD
016724  000000                          0                       ;THIRD DATA WORD
016726  000000                          0                       ;FOURTH DATA WORD

                                        ;*FILL WRITE FROM BUFFER WITH DATA

016730  004037  033164                  JSR     R0,@#CLAREA     ;CLEAR 256. WORDS, FROM WRFROM+10
016734  002400                          WRFROM+10                       ;STARTING FROM WRFROM+10
```

```
4888  016736  000400                  256.                    ;256. WORDS
4889  016740  000000                  0                       ;FILL WITH 0
4890
4891
4892                                  ;*NOW READ INTO BUFFER WILL BE FILLED WITH SAME DATA
4893                                  ;*AS WRITE FROM BUFFER SO THAT AFTER A WRITE COMPARISONS
4894                                  ;*CAN BE MADE TO MAKE SURE THAT WRITE DID NOT
4895                                  ;*CHANGE WRITE FROM BUFFER
4896
4897
4898  016742  004037  033140          JSR     R0,@#FLHEAD     ;SAVE HEADER DATA IN REINTO
4899  016746  003434                  REINTO                  ;LOCATION WHERE SAVED
4900  016750  000004                  4                       ;NUMBER OF WORDS SAVED
4901  016752  010000                  10000                   ;FIRST DATA WORD
4902  016754  000001                  1                       ;SECOND DATA WORD
4903  016756  000000                  0                       ;THIRD DATA WORD
4904  016760  000000                  0                       ;FOURTH DATA WORD
4905
4906  016762  004037  033164          JSR     R0,@#CLAREA     ;CLEAR 256. WORDS, FROM REINTO+10
4907  016766  003444                  REINTO+10               ;       ;STARTING FROM REINTO+10
4908  016770  000400                  256.                    ;256. WORDS
4909  016772  000000                  0                       ;FILL WITH 0
4910
4911
4912                                  ;*NOW THE WRITE HEADER AND DATA COMMAND WILL BE FILLED
4913
4914  016774  004037  035276          JSR     R0,@#RUN        ;SETUP TO RUN FOR DATA COMMAND
4915  017000  000000                  0                       ;CYLINDER 0
4916  017002     001                  .BYTE   1               ;SECTOR 1
4917  017003     000                  .BYTE   0               ;TRACK 0
4918  017004  177374                  -256.-4                 ;WORD COUNT (DATA) = 256. +
4919                                                          ;4 HEADER WORDS
4920  017006  002370                  WRFROM                  ;BUS ADDRESS
4921                                                          ;STARTING ADDRESS OF DATA
4922                                                          ;BUFFER = WRFROM
4923  017010  000000                  0                       ;DO NOT INHIBIT BUS ADDRESS INCREMENT
4924  017012  010000                  FMT22                   ;16 BITS PER WORD FORMAT
4925                                                          ;DO NOT INHIBIT ECC CORRECTION
4926                                                          ;DO NOT INHIBIT HEADER COMPARE
4927  017014  002344                  WRIFOR                  ;GET READY TO DO A WRIFOR
4928                                                  ;WRITE HEADER AND DATA WITH 62 IN RHCS1
4929
4930
4931                                  ;*NOW SAVE REGISTERS FOR COMPARISON AFTER WRITE HEADER AND DATA
4932  017016  004037  033462          JSR     R0,@#SAVER      ;SAVE REGISTERS
4933  017022  002172                  RHWC                    ;RHWC IS THE FIRST REGISTER SAVED
4934  017024  004512                  SAVERE                  ;STARTING ADDRES OF WHERE
4935                                                          ;THE REGISTERS ARE SAVED
4936  017026  000021                  17.                     ;NUMBER OF REGISTERS
4937                                                          ;SAVED = 17.
4938
4939  017030  004737  033374          JSR     PC,@#CHECKT     ;CHECKS DVA,RDY,MOL,DPR,DRY AND VV = 1
4940                                                          ;AND THAT ALL STATUS BITS ARE = 0
4941  017034  104401  057465          TYPE    ,CPHALT         ;CANNOT CONTINUE TESTING IF NOT
4942  017040  000000                  HALT                    ;STOP TEST
4943
```

RESTORE SECTOR 1 CYLINDER 1 TRACK 1 SEQ 0106

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

                                        MOV     @#WRIFOR,-(SP)  ;GET READY TO MOVE COMMAND
                                        BIS     #GO!IE,(SP)     ;GET READY TO SET GO AND
                                                                ;ENABLE INTERRUPT
                                        MOV     (SP)+,@RHCS1    ;GO WITH
                                                                 ;62 IN RHCS1 FOR WRITE HEADER AND DATA
                                                                ;WITH INTERRUPT ENABLED
                                        MOV     @R1,R0          ;SAVE RHCS1 DURING ABOVE OPERATION
                                        MOV     @R3,R5          ;SAVE RHDS1 DURING ABOVE OPERATION
                                        ;*ONE REVOLUTION=16670 MICRO SEC, ONE SECTOR = 760 MICRO SEC

                                        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 IS ISSUED

                                        MOV     @#WRIFOR,-(SP)  ;SAVE COMMAND
                                        BIS     #IE!GO!DVA,(SP) ;INCLUDE IE!GO!DVA
                                        MOV     (SP),@#$GDDAT   ;SAVE FOR PRINTOUT
                                        CMP     (SP)+,R0        ;DURING ABOVE OPERATION ONLY IE!GO!DVA
                                                                ;AND COMMAND SHOULD BE SET
                                        BEQ     64$             ;BRANCH IF GOOD
                                        MOV     R0,@#$BDDAT     ;BAD DATA
                                        MOV     R1,@#REGADR     ;FAILING REGISTER RHCS1
                                        ERROR   21              ;DURING ABOVE OPERATION ONLY
                                                                ;COMMAND AND IE!GO!DVA SHOULD BE SET
                                64$:    MOV     #MOL!DPR!VV,-(SP)       ;SAVE BITS SET DURING OPERATION IN RHDS1
                                        MOV     (SP),@#$GDDAT   ;SAVE FOR PRINTOUT
                                        CMP     (SP)+,R5        ;DURING ABOVE OPERATION ONLY MOL!DPR!VV
                                                                ;SHOULD BE SET
                                        BEQ     66$             ;BRANCH IF GOOD
                                        MOV     R5,@#$BDDAT     ;BAD DATA
                                        MOV     R3,@#REGADR     ;FAILING REGISTER RHDS1
                                        ERROR   63              ;DURING ABOVE OPERATION ONLY
                                                                ;MOL!DPR!VV SHOULD BE SET
                                66$:

                                        ;*NOW CHANGE SAVED REGISTERS TO EXPECTED VALUES

                                        JSR     R0,@#FILLRE     ;MOV 0 INTO SAVED RHWC
                                        RHWC                    ;SAVED REGISTER TO CHANGE
                                        0                       ;DATA
```

```
5000  017172  004037  033216                JSR     R0,@#FILLRE     ;MOV WRFROM+<260.*2> INTO SAVED RHBA
5001  017176  002174                        RHBA                    ;SAVED REGISTER TO CHANGE
5002  017200  003400                        WRFROM+<260.*2>                 ;DATA
5003
5004
5005  017202  004037  033216                JSR     R0,@#FILLRE     ;MOV 2 INTO SAVED RHDST
5006  017206  002204                        RHDST                   ;SAVED REGISTER TO CHANGE
5007  017210  000002                        2                       ;DATA
5008
5009
5010                                        ;*NOW COMPARE REGISTERS BEFORE WRITE HEADER AND DATA
5011                                        ;*WITH REGISTERS AFTER COMMAND
5012
5013
5014  017212  004037  034312                JSR     R0,@#COMREG     ;COMPARE SAVED REGISTERS WITH
5015                                                                ;PRESENT VALUE
5016  017216  004512                        SAVERE                  ;GOOD DATA SAVED IN 'SAVERE'
5017  017220  002254                        WC                      ;TEST DATA STARTING FROM 'RHWC'
5018  017222  000021                        17.                     ;17. REGISTERS TO BE COMPARED
5019  017224  017230                        1$                      ;RETURN TO 1$ ON ERROR
5020  017226  017234                        2$                      ;RETURN TO 2$ ON NO ERROR
5021
5022
5023  017230  104027              1$:       ERROR   27              ;WRITE HEADER AND DATA
5024  017232  000207                        RTS     PC              ;CAUSED IMPROPER REGISTER
5025                                                                ;CHANGE
5026                                                                ;GOOD DATA GIVES WHAT SHOULD
5027                                                                ;BE THERE
5028                                                                ;RECEIVED DATA GIVES WHAT
5029                                                                ;WAS THERE AFTER COMMAND
5030
5031                                        ;*NOW WRITE FROM BUFFER WILL BE CHECKED TO SEE THAT
5032                                        ;*NOTHER GOT CHANGED
5033  017234                        2$:
5034
5035  017234  004037  035342                JSR     R0,@#COMPAR     ;COMPARE TWO BLOCKS OF MEMORY
5036  017240  003434                        REINTO                  ;GOOD DATA STARTS FROM REINTO
5037  017242  002370                        WRFROM                  ;TEST DATA STARTS FROM WRFROM
5038  017244  000404                        260.                    ;260., WORDS TO BE COMPARED
5039  017246  017252                        3$                      ;RETURN TO 3$ ON ERROR
5040  017250  017256                        4$                      ;RETURN TO 4$ ON NO ERROR
5041
5042
5043  017252  104030              3$:       ERROR   30              ;WRITE HEADER AND DATA
5044  017254  000207                        RTS     PC              ;CHANGED WRITE FROM BUFFER
5045
5046                                        ;*NOW A READ HEADER AND DATA COMMAND WILL BE GIVEN
5047                                        ;*READ INTO BUFFER IS FILLED WITH ONES
5048  017256                        4$:
5049
5050  017256  004737  033314                JSR     PC,@#CLDISK     ;SET R1-RHCS1, R2-RHCS2
5051                                                                ;R3-RHDS1, R4-RHER1
5052                                                                ;GIVE RH-11 INITIALIZE
5053                                                                ;SETUP UNIT NUMBER
5054
5055  017262  004037  033164                JSR     R0,@#CLAREA     ;CLEAR 260. WORDS, FROM REINTO
```

```
5056  017266  003434                          REINTO                  ;STARTING FROM REINTO
5057  017270  000404                          260.                    ;260. WORDS
5058  017272  177777                          -1                      ;FILL WITH -1
5059
5060
5061                                          ;*NOW FILL COMMAND
5062
5063  017274  004037  035276                  JSR     R0,@#RUN        ;SETUP TO RUN FOR DATA COMMAND
5064  017300  000000                          0                       ;CYLINDER 0
5065  017302     001                          .BYTE   1               ;SECTOR 1
5066  017303     000                          .BYTE   0               ;TRACK 0
5067  017304  177374                          -256.-4                 ;WORD COUNT (DATA) = 256. +
5068                                                                  ;4 HEADER WORDS
5069  017306  003434                          REINTO                  ;BUS ADDRESS
5070                                                                  ;STARTING ADDRESS OF DATA
5071                                                                  ;BUFFER = REINTO
5072  017310  000000                          0                       ;DO NOT INHIBIT BUS ADDRESS INCREMENT
5073  017312  014000                          EC1!FMT22                       ;16 BITS PER WORD FORMAT
5074                                                                  ;INHIBIT ECC CORRECTION
5075                                                                  ;DO NOT INHIBIT HEADER COMPARE
5076  017314  002350                          REFOR                   ;GET READY TO DO A REFOR
5077                                                                  ;READ HEADER AND DATA WITH 72 IN RHCS1
5078
5079
5080                                          ;*NOW SAVE REGISTERS FOR COMPARISON AFTER READ HEADER AND DATA
5081  017316  004037  033462                  JSR     R0,@#SAVER      ;SAVE REGISTERS
5082  017322  002172                          RHWC                    ;RHWC IS THE FIRST REGISTER SAVED
5083  017324  004512                          SAVERE                  ;STARTING ADDRES OF WHERE
5084                                                                  ;THE REGISTERS ARE SAVED
5085  017326  000022                          18.                     ;NUMBER OF REGISTERS
5086                                                                  ;SAVED = 18.
5087
5088  017330  004737  033374                  JSR     PC,@#CHECKT     ;CHECKS DVA,RDY,MOL,DPR,DRY AND VV = 1
5089                                                                  ;AND THAT ALL STATUS BITS ARE = 0
5090  017334  104401  057465                  TYPE    ,CPHALT         ;CANNOT CONTINUE TESTING IF NOT
5091  017340  000000                          HALT                    ;STOP TEST
5092
5093  017342  013777  004506  162616          MOV     @#RP4VEC,@RPVEC ;SET RP04 VECTOR ADDRESS
5094                                                                  ;TO 'TIME1' IF P-CLOCK IS PRESENT
5095                                                                  ;OR TO 'TIME2' IF P-CLOCK IS NOT THERE
5096                                                                  ;'TIME' WILL ONLY SAVE
5097                                                                  ;CURRENT CYLINDER ADDRESS
5098                                                                  ;AND LOOK AHEAD REGISTERS
5099
5100
5101  017350  013746  002350                  MOV     @#REFOR,-(SP)   ;GET READY TO MOVE COMMAND
5102  017354  052716  000101                  BIS     #GO!IE,(SP)     ;GET READY TO SET GO AND
5103                                                                  ;ENABLE INTERRUPT
5104  017360  012677  162614                  MOV     (SP)+,@RHCS1    ;GO WITH
5105                                                                    ;72 IN RHCS1 FOR READ DATA
5106                                                                  ;WITH INTERRUPT ENABLED
5107  017364  011100                          MOV     @R1,R0          ;SAVE RHCS1 DURING ABOVE OPERATION
5108  017366  011305                          MOV     @R3,R5          ;SAVE RHDS1 DURING ABOVE OPERATION
5109
5110
5111  017370  104413                          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 IS ISSUED

                                        MOV     @#REFOR,-(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,@#$BDDAT     ;BAD DATA
                                        MOV     R3,@#REGADR     ;FAILING REGISTER RHDS1
                                        ERROR   63              ;DURING ABOVE OPERATION ONLY
                                                                ;MOL!DPR!VV SHOULD BE SET
                                69$:

                                        ;*CHEANGE 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+<260.*2> INTO SAVED RHBA
                                        RHBA                    ;SAVED REGISTER TO CHANGE
                                        REINTO+<260.*2>                 ;DATA

                                        JSR     R0,@#FILLRE     ;MOV 2 INTO SAVED RHDST
                                        RHDST                   ;SAVED REGISTER TO CHANGE
                                        2                       ;DATA


                                        ;*COMPARE REGISTER BEFORE READ HEADER AND DATA
                                        ;*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   31              ;READ HEADER AND DATA CAUSED
        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

6$:

        JSR     R0,@#COMPAR     ;COMPARE TWO BLOCKS OF MEMORY
        WRFROM                  ;GOOD DATA STARTS FROM WRFROM
        REINTO                  ;TEST DATA STARTS FROM REINTO
        260.                    ;260., WORDS TO BE COMPARED
        7$                      ;RETURN TO 7$ ON ERROR
        10$                     ;RETURN TO 10$ ON NO ERROR

7$:     ERROR   32              ;WRITE HEADER AND DATA
        RTS     PC              ;FOLLOWED BY A READ HEADER
                                ;AND DATA GAVE A READ ERROR
                                ;ERROR MAY BE IN READ OR WRITE

10$:
```

```
                                        ;;*****************************************************************
                                        ;*TEST 20        INVALID ADDRESS ERROR - RHER1 - 'IAE'

                                        ;*      A WRITE HEADER AND DATA WILL BE ATTEMPTED TO CYLINDER 411./815.
                                        ;*      TRACK 0, SECTOR 0

                                        ;*      INVALID ADDRESS ERROR (IAE) BIT #10 IN RHER1 SHOULD SET

                                        ;;*****************************************************************
017556  000004                        IST20:  SCOPE
017560  012706  001000                        MOV     #STACK,SP       ;RESET STACK
017564  012737  000020  004504                MOV     #20,@#TSTNM     ;SAVE TEST NUMBER

017572  004737  033314                        JSR     PC,@#CLDISK     ;SET R1-RHCS1, R2-RHCS2
                                                                        ;R3-RHDS1, R4-RHER1
                                                                        ;GIVE RH-11 INITIALIZE
                                                                        ;SETUP UNIT NUMBER


                                        ;*CHECK THE DRIVE TYPE AND THEN FILL THE
                                        ;*WRITE FROM BUFFER WITH APPROPRIATE HEADER

                                        ;;*****************************************************************
017576  005737  004636                        TST     @#RP06  ;TEST FOR RP06 DRIVE
017602  001411                                BEQ     5$              ;TREAT UNIT AS AN RP04
                                                                        ;TREAT AS AN RP06
                                        ;;*****************************************************************


017604  004037  033140                        JSR     R0,@#FLHEAD     ;SAVE HEADER DATA IN WRFROM
017610  002370                        WRFROM                          ;LOCATION WHERE SAVED
017612  000004                        4                               ;NUMBER OF WORDS SAVED
017614  011457                        11457                           ;FIRST DATA WORD
017616  000000                        0                               ;SECOND DATA WORD
017620  000000                        0                               ;THIRD DATA WORD
017622  000000                        0                               ;FOURTH DATA WORD
017624  000410                        BR      6$              ;CONTINUE WITH SET UP

017626                          5$:

017626  004037  033140                        JSR     R0,@#FLHEAD     ;SAVE HEADER DATA IN WRFROM
017632  002370                        WRFROM                          ;LOCATION WHERE SAVED
017634  000004                        4                               ;NUMBER OF WORDS SAVED
017636  010633                        10633                           ;FIRST DATA WORD
017640  000000                        0                               ;SECOND DATA WORD
017642  000000                        0                               ;THIRD DATA WORD
017644  000000                        0                               ;FOURTH DATA WORD
017646                          6$:                             ;CONTINUE
                                        ;*FILL WRITE FROM BUFFER WITH DATA


017646  004037  033164                        JSR     R0,@#CLAREA     ;CLEAR 256. WORDS, FROM WRFROM+10
017652  002400                        WRFROM+10                               ;STARTING FROM WRFROM+10
017654  000400                        256.                            ;256. WORDS
017656  177777                        0-1                             ;FILL WITH 0-1
```

```
                                        ;*THE DRIVE TYPE IS CHECKED AND THE APPROPRIATE
                                        ;*WRITE HEADER AND DATA COMMAND IS LOADED


                                    ;;*****************************************************************
017660  005737  004636                TST     @#RP06  ;TEST FOR RP06 DRIVE
017664  001412                        BEQ     3$              ;TREAT UNIT AS RP04
                                                            ;TREAT UNIT AS RP06
                                    ;;*****************************************************************


017666  004037  035276                JSR     R0,@#RUN        ;SETUP TO RUN FOR DATA COMMAND
017672  001457                        815.                    ;CYLINDER 815.
017674     000                      .BYTE   0               ;SECTOR 0
017675     000                      .BYTE   0               ;TRACK 0
017676  177374                      -256.-4                 ;WORD COUNT (DATA) = 256. +
                                                            ;4 HEADER WORDS
017700  002370                      WRFROM                  ;BUS ADDRESS
                                                            ;STARTING ADDRESS OF DATA
                                                            ;BUFFER = WRFROM
017702  000000                      0                       ;DO NOT INHIBIT BUS ADDRESS INCREMENT
017704  010000                      FMT22                   ;16 BITS PER WORD FORMAT
                                                            ;DO NOT INHIBIT ECC CORRECTION
                                                            ;DO NOT INHIBIT HEADER COMPARE
017706  002344                      WRIFOR                  ;GET READY TO DO A WRIFOR
                                                    ;WRITE HEADER AND DATA WITH 62 IN RHCS1

017710  000411                      BR      4$              ;CONTINUE WITH TESTING
017712                          3$:

017712  004037  035276                JSR     R0,@#RUN        ;SETUP TO RUN FOR DATA COMMAND
017716  000633                      411.                    ;CYLINDER 411.
017720     000                      .BYTE   0               ;SECTOR 0
017721     000                      .BYTE   0               ;TRACK 0
017722  177374                      -256.-4                 ;WORD COUNT (DATA) = 256. +
                                                            ;4 HEADER WORDS
017724  002370                      WRFROM                  ;BUS ADDRESS
                                                            ;STARTING ADDRESS OF DATA
                                                            ;BUFFER = WRFROM
017726  000000                      0                       ;DO NOT INHIBIT BUS ADDRESS INCREMENT
017730  010000                      FMT22                   ;16 BITS PER WORD FORMAT
                                                            ;DO NOT INHIBIT ECC CORRECTION
                                                            ;DO NOT INHIBIT HEADER COMPARE
017732  002344                      WRIFOR                  ;GET READY TO DO A WRIFOR
                                                    ;WRITE HEADER AND DATA WITH 62 IN RHCS1

017734                          4$:                         ;CONTINUE


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

017734  004037  033462                JSR     R0,@#SAVER      ;SAVE REGISTERS
```

```
                                        RHWC                    ;RHWC IS THE FIRST REGISTER SAVED
                                        SAVERE                  ;STARTING ADDRES OF WHERE
                                                                ;THE REGISTERS ARE SAVED
                                        18.                     ;NUMBER OF REGISTERS
                                                                ;SAVED = 18.

                                        JSR     PC,@#CHECKT     ;CHECKS DVA,RDY,MOL,DPR,DRY AND VV = 1
                                                                ;AND THAT ALL STATUS BITS ARE = 0
                                        TYPE    ,CPHALT         ;CANNOT CONTINUE TESTING IF NOT
                                        HALT                    ;STOP TEST

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


                                        MOV     @#WRIFOR,-(SP)  ;GET READY TO MOVE COMMAND
                                        BIS     #GO!IE,(SP)     ;GET READY TO SET GO AND
                                                                ;ENABLE INTERRUPT
                                        MOV     (SP)+,@RHCS1    ;GO WITH
                                                                 ;62 IN RHCS1 FOR WRITE HEADER AND DATA
                                                                ;WITH INTERRUPT ENABLED


                                        WAT                     ;WAIT FOR IAE BIT TO SET
                                        RHER1                   ;WAIT FOR RHER1 REGISTER
                                        IAE                     ;WAIT FOR IAE BIT IN RHER1 REGISTER
                                        9.                      ;ALLOW 90 MICRO SECONDS
                                        9.                      ;IAE MUST SET BETWEEN
                                                                ;00 AND 180 MICRO SECONDS



                                        ;*CHANGE        THE SAVED REGISTERS TO EXPECTED VALUES

                                        ;*AS EXCEPTION IS ASSERTED BEFORE RUN IS LATCHED,
                                        ;*RHWC,RHBA,RHCS1 & RHCS2 CANNOT BE PREDETERMINED -
                                        ;*THEY WILL VARY DEPENDING ON GATE DELAYS FOR DIFFERENT UNITS

                                        MOV     @RHWC,@#SAVERE  ;RHWC IS UNPREDICTABLE
                                                                ;AS EXPLAINED ABOVE
                                        MOV     @RHBA,@#SAVERE+2;RHBA IS UNPREDICTABLE
                                                                ;AS EXPLAINED ABOVE
                                        MOV     @RHCS2,@#SAVERE+4;RHCS2 IS UNPREDICTABLE
                                                                ;AS EXPLAINED ABOVE
                                        MOV     @RHCS1,@#SAVERE+6;RHCS1 IS UNPREDICTABLE
                                                                ;AS EXPLAINED ABOVE

                                        JSR     R0,@#CHREG      ;CHANGE BITS IN SAVED REGISTER
                                        RHDS1                   ;CHANGE RHDS1 REGISTER

                                        2                       ;2 BIT/BITS TO BE CHANGED
                                        1                       ;NEW VALUE OF ATA IS 1
```

```
5368  020056  100000                          ATA                     ;CHANGE ATA BIT
5369  020060  000001                          1                       ;NEW VALUE OF ERR IS 1
5370  020062  040000                          ERR                     ;CHANGE ERR BIT
5371
5372  020064  004037  034204                  JSR     R0,@#CHREG      ;CHANGE BITS IN SAVED REGISTER
5373  020070  002202                          RHER1                   ;CHANGE RHER1 REGISTER
5374
5375  020072  000001                          1                       ;1 BIT/BITS TO BE CHANGED
5376  020074  000001                          1                       ;NEW VALUE OF IAE IS 1
5377  020076  002000                          IAE                     ;CHANGE IAE BIT
5378
5379  020100  053737  004644  004536          BIS     @#ATTENT,@#SAVERE+24 ;SET APPROPIATE 'ATA' BITS
5380                                                                  ;FOR WORKING DRIVE IN
5381                                                                  ;SAVED RHAS LOCATION
5382  020106  017737  162072  004524          MOV     @RHDST,@#SAVERE+12 ;RHDST IS INDETERMINATE
5383                                                                  ;SO IT IS NOT CHECKED
5384
5385
5386                                          ;*COMPARE REGISTERS BEFORE ATTEMPTED WRITE WITH
5387                                          ;*CONTENTS AFTER ATTEMPTED WRITE WITH AN 'IAE' ERROR
5388                                  ;;**************************************************************
5389
5390
5391  020114  004037  034312                  JSR     R0,@#COMREG     ;COMPARE SAVED REGISTERS WITH
5392                                                                  ;PRESENT VALUE
5393  020120  004512                          SAVERE                  ;GOOD DATA SAVED IN 'SAVERE'
5394  020122  002254                          WC                      ;TEST DATA STARTING FROM 'RHWC'
5395  020124  000022                          18.                     ;18. REGISTERS TO BE COMPARED
5396  020126  020132                          1$                      ;RETURN TO 1$ ON ERROR
5397  020130  020136                          2$                      ;RETURN TO 2$ ON NO ERROR
5398
5399
5400  020132  104054                  1$:     ERROR   54              ;ATTEMPTED OPERATION WITH
5401  020134  000207                          RTS     PC              ;INVALID ADDRESS CAUSED
5402                                                                  ;IMPROPER REGISTER CHANGE
5403                                                                  ;GOOD DATA GIVES WHAT SHOULD
5404                                                                  ;BE THERE
5405  020136                          2$:                             ;RECEIVED DATA GIVES REGISTER
5406                                                                  ;CONTENTS AFTER ATTEMPTED
5407                                                                  ;WRITE HEADER AND DATA
5408
```

```
                                        ;;*****************************************************************
                                        ;*TEST 21        INVALID ADDRESS ERROR - RHER1 (BIT #10)IAE

                                        ;*      A WRITE DATA IS ATTEMPTED TO CYLINDER 0, TRACK 19,
                                        ;*      SECTOR 0
                                        ;*      INVALID ADDRESS ERROR IAE BIT #10 IN RHER1
                                        ;*      SHOULD SET

                                        ;;*****************************************************************
020136  000004                  ST21:   SCOPE
020140  012706  001000                  MOV     #STACK,SP       ;RESET STACK
020144  012737  000021  004504          MOV     #21,@#TSTNM     ;SAVE TEST NUMBER

020152  004737  033314                  JSR     PC,@#CLDISK     ;SET R1-RHCS1, R2-RHCS2
                                                                ;R3-RHDS1, R4-RHER1
                                                                ;GIVE RH-11 INITIALIZE
                                                                ;SETUP UNIT NUMBER

                                        ;*FILL WRITE FROM BUFFER WITH DATA

020156  004037  033164                  JSR     R0,@#CLAREA     ;CLEAR 256. WORDS, FROM WRFROM
020162  002370                          WRFROM                  ;STARTING FROM WRFROM
020164  000400                          256.                    ;256. WORDS
020166  000377                          377                     ;FILL WITH 377


                                        ;*WRITE DATA COMMAND WILL BE FILLED

020170  004037  035276                  JSR     R0,@#RUN        ;SETUP TO RUN FOR DATA COMMAND
020174  000000                          0                       ;CYLINDER 0
020176     000                          .BYTE   0               ;SECTOR 0
020177     023                          .BYTE   19.             ;TRACK 19.
020200  177400                          -256.                   ;WORD COUNT = 256.
020202  002370                          WRFROM                  ;BUS ADDRESS
                                                                ;STARTING ADDRESS OF DATA
                                                                ;BUFFER = WRFROM
020204  000000                          0                       ;DO NOT INHIBIT BUS ADDRESS INCREMENT
020206  010000                          FMT22                   ;16 BITS PER WORD FORMAT
                                                                ;DO NOT INHIBIT ECC CORRECTION
                                                                ;DO NOT INHIBIT HEADER COMPARE
020210  002342                          WRIDAT                  ;GET READY TO DO A WRIDAT
                                                        ;WRITE DATA WITH 60 IN RHCS1


                                        ;*SAVE REGISTERS FOR COMPARISON AFTER ATTEMPTED WRITE DATA
020212  004037  033462                  JSR     R0,@#SAVER      ;SAVE REGISTERS
020216  002172                          RHWC                    ;RHWC IS THE FIRST REGISTER SAVED
020220  004512                          SAVERE                  ;STARTING ADDRES OF WHERE
                                                                ;THE REGISTERS ARE SAVED
020222  000022                          18.                     ;NUMBER OF REGISTERS
                                                                ;SAVED = 18.

020224  004737  033374                  JSR     PC,@#CHECKT     ;CHECKS DVA,RDY,MOL,DPR,DRY AND VV = 1
                                                                ;AND THAT ALL STATUS BITS ARE = 0
020230  104401  057465                  TYPE    ,CPHALT         ;CANNOT CONTINUE TESTING IF NOT
```

```
5465  020234  000000                          HALT                    ;STOP TEST
5466
5467  020236  013777  004506  161722        MOV     @#RP4VEC,@RPVEC ;SET RP04 VECTOR ADDRESS
5468                                                                  ;TO 'TIME1' IF P-CLOCK IS PRESENT
5469                                                                  ;OR TO 'TIME2' IF P-CLOCK IS NOT THERE
5470                                                                  ;'TIME' WILL ONLY SAVE
5471                                                                  ;CURRENT CYLINDER ADDRESS
5472                                                                  ;AND LOOK AHEAD REGISTERS
5473
5474
5475  020244  013746  002342                MOV     @#WRIDAT,-(SP)  ;GET READY TO MOVE COMMAND
5476  020250  052716  000101                BIS     #GO!IE,(SP)     ;GET READY TO SET GO AND
5477                                                                  ;ENABLE INTERRUPT
5478  020254  012677  161720                MOV     (SP)+,@RHCS1    ;GO WITH
5479                                                                   ;60 IN RHCS1 FOR WRITE DATA
5480                                                                  ;WITH INTERRUPT ENABLED
5481
5482
5483  020260  104413                         WAT                     ;WAIT FOR IAE BIT TO SET
5484  020262  002202                         RHER1                   ;WAIT FOR RHER1 REGISTER
5485  020264  002000                         IAE                     ;WAIT FOR IAE BIT IN RHER1 REGISTER
5486  020266  000011                         9.                      ;ALLOW 90 MICRO SECONDS
5487  020270  000011                         9.                      ;IAE MUST SET BETWEEN
5488                                                                  ;00 AND 180 MICRO SECONDS
5489
5490                                         ;*CHANGE        SAVED REGISTERS TO EXPECTED VALUES
5491
5492
5493
5494                                         ;*AS EXCEPTION IS ASSERTED BEFORE RUN IS LATCHED
5495                                         ;*RHWC,RHBA,RHCS1,RHCS2,  CANNOT BE PEREDETERMINED
5496                                         ;*THEY WILL VARY DEPENDING ON GATE DELAYS ON DIFFRENT UNITS
5497
5498  020272  017737  161674  004512        MOV     @RHWC,@#SAVERE  ;RHWC IS UNPREDICTABLE
5499                                                                  ;AS EXPLAINED ABOVE
5500  020300  017737  161670  004514        MOV     @RHBA,@#SAVERE+2;RHBA IS UNPREDICTABLE
5501                                                                  ;AS EXPLAINED ABOVE
5502  020306  017737  161664  004516        MOV     @RHCS2,@#SAVERE+4;RHCS2 IS UNPREDICTABLE
5503                                                                  ;AS EXPLAINED ABOVE
5504  020314  017737  161660  004520        MOV     @RHCS1,@#SAVERE+6;RHCS1 IS UNPREDICTABLE
5505                                                                  ;AS EXPLAINED ABOVE
5506
5507  020322  004037  034204                JSR     R0,@#CHREG      ;CHANGE BITS IN SAVED REGISTER
5508  020326  002222                         RHDS1                   ;CHANGE RHDS1 REGISTER
5509
5510  020330  000002                         2                       ;2 BIT/BITS TO BE CHANGED
5511  020332  000001                         1                       ;NEW VALUE OF ATA IS 1
5512  020334  100000                         ATA                     ;CHANGE ATA BIT
5513  020336  000001                         1                       ;NEW VALUE OF ERR IS 1
5514  020340  040000                         ERR                     ;CHANGE ERR BIT
5515  020342  017737  161636  004524        MOV     @RHDST,@#SAVERE+12 ;RHDST IS INDETERMINATE SO IT IS NOT CHECKED
5516
5517
5518  020350  004037  034204                JSR     R0,@#CHREG      ;CHANGE BITS IN SAVED REGISTER
5519  020354  002202                         RHER1                   ;CHANGE RHER1 REGISTER
5520
```

```
5521  020356  000001                          1                       ;1 BIT/BITS TO BE CHANGED
5522  020360  000001                          1                       ;NEW VALUE OF IAE IS 1
5523  020362  002000                          IAE                     ;CHANGE IAE BIT
5524
5525  020364  053737  004644  004536          BIS     @#ATTENT,@#SAVERE+24 ;SET APPROPIATE 'ATA' BITS
5526                                                                    ;FOR WORKING DRIVE IN
5527                                                                    ;SAVED RHAS LOCATION
5528
5529                                          ;*COMPARE REGISTERS BEFORE ATTEMPTED WRITE DATA
5530                                          ;*WITH AFTER ATTEMPT, IAE SHOULD BE SET
5531
5532  020372  004037  034312                  JSR     R0,@#COMREG     ;COMPARE SAVED REGISTERS WITH
5533                                                                    ;PRESENT VALUE
5534  020376  004512                          SAVERE                  ;GOOD DATA SAVED IN 'SAVERE'
5535  020400  002254                          WC                      ;TEST DATA STARTING FROM 'RHWC'
5536  020402  000022                          18.                     ;18. REGISTERS TO BE COMPARED
5537  020404  020410                          1$                      ;RETURN TO 1$ ON ERROR
5538  020406  020414                          2$                      ;RETURN TO 2$ ON NO ERROR
5539
5540
5541  020410  104054                  1$:     ERROR   54              ;ATTEMPTED WRITE DATA
5542  020412  000207                          RTS     PC              ;WITH INVALID ADDRESS
5543                                                                    ;CAUSED IMPROPER REGISTER
5544                                                                    ;CHANGE
5545  020414                          2$:                             ;GOOD DATA GIVES WHAT
5546                                                                    ;SHOULD BE THERE
5547                                                                    ;RECEIVED DATA GIVES WHAT
5548                                                                    ;WAS THERE AFTER AFTER ATTEMPT
5549
```

```
                                ;;*****************************************************************
                                ;*TEST 22       INVALID ADDRESS ERROR RHER1 -BIT #10

                                ;*      A READ HEADER AND DATA IS ATTEMPTED TO CYLINDER 0
                                ;*      TRACK 0, SECTOR 22
                                ;*      INVALID ADDRESS ERROR IAE BIT #10 IN RHER1
                                ;*      SHOULD SET
                                ;*      THIS WILL START WITH THE HEADS ON CYLINDER 10
                                ;*      TO PROVE THAT IAE SETS EVEN BEFORE THE IMPLIED
                                ;*      SEEK

                                ;;*****************************************************************
020414  000004                  TST22:  SCOPE
020416  012706  001000                  MOV     #STACK,SP       ;RESET STACK
020422  012737  000022  004504          MOV     #22,@#TSTNM     ;SAVE TEST NUMBER

020430  004737  033314                  JSR     PC,@#CLDISK     ;SET R1-RHCS1, R2-RHCS2
                                                                ;R3-RHDS1, R4-RHER1
                                                                ;GIVE RH-11 INITIALIZE
                                                                ;SETUP UNIT NUMBER

                                        ;*GET THE HEADS TO CYLINDER 10

020434  004737  033374                  JSR     PC,@#CHECKT     ;CHECKS DVA,RDY,MOL,DPR,DRY AND VV = 1
                                                                ;AND THAT ALL STATUS BITS ARE = 0
020440  104401  057465                  TYPE    ,CPHALT         ;CANNOT CONTINUE TESTING IF NOT
020444  000000                          HALT                    ;STOP TEST

020446  013777  004506  161512          MOV     @#RP4VEC,@RPVEC ;SET RP04 VECTOR ADDRESS
                                                                ;TO 'TIME1' IF P-CLOCK IS PRESENT
                                                                ;OR TO 'TIME2' IF P-CLOCK IS NOT THERE
                                                                ;'TIME' WILL ONLY SAVE
                                                                ;CURRENT CYLINDER ADDRESS
                                                                ;AND LOOK AHEAD REGISTERS


020454  004037  033264                  JSR     R0,@#SEEKCY     ;SEEK FOR
020460  000010                          10                      ;CYLINDER 10


020462  013746  002352                  MOV     @#SEECOM,-(SP)  ;GET READY TO MOVE COMMAND
020466  052716  000101                  BIS     #GO!IE,(SP)     ;GET READY TO SET GO AND
                                                                ;ENABLE INTERRUPT
020472  012677  161502                  MOV     (SP)+,@RHCS1    ;GO WITH
                                                                 ;4 IN RHCS1 FOR SEEK
                                                                ;WITH INTERRUPT ENABLED


020476  104413                          WAT                     ;WAIT FOR DRY BIT TO SET
020500  002222                          RHDS1                   ;WAIT FOR RHDS1 REGISTER
020502  000200                          DRY                     ;WAIT FOR DRY BIT IN RHDS1 REGISTER
020504  015530                          7000.                   ;ALLOW 70000 MICRO SECONDS
020506  015530                          7000.                   ;DRY MUST SET BETWEEN
                                                                ;00 AND 140000 MICRO SECONDS
```

```
5606
5607
5608  020510  004737  033314                JSR     PC,@#CLDISK     ;SET R1-RHCS1, R2-RHCS2
5609                                                                  ;R3-RHDS1, R4-RHER1
5610                                                                  ;GIVE RH-11 INITIALIZE
5611                                                                  ;SETUP UNIT NUMBER
5612
5613                                        ;*FILL READ INTO BUFFER WITH 125252
5614
5615  020514  004037  033164                JSR     R0,@#CLAREA     ;CLEAR 260 WORDS, FROM REINTO
5616  020520  003434                        REINTO                  ;STARTING FROM REINTO
5617  020522  000260                        260                     ;260 WORDS
5618  020524  125252                        125252                  ;FILL WITH 125252
5619
5620
5621                                        ;*THE READ HEADER AND DATA COMMAND IS FILLED
5622
5623  020526  004037  035276                JSR     R0,@#RUN        ;SETUP TO RUN FOR DATA COMMAND
5624  020532  000000                        0                       ;CYLINDER 0
5625  020534     026                        .BYTE   22.             ;SECTOR 22.
5626  020535     000                        .BYTE   0               ;TRACK 0
5627  020536  177374                        -256.-4                 ;WORD COUNT (DATA) = 256. +
5628                                                                  ;4 HEADER WORDS
5629  020540  003434                        REINTO                  ;BUS ADDRESS
5630                                                                  ;STARTING ADDRESS OF DATA
5631                                                                  ;BUFFER = REINTO
5632  020542  000000                        0                       ;DO NOT INHIBIT BUS ADDRESS INCREMENT
5633  020544  014000                        FMT22!ECI                       ;16 BITS PER WORD FORMAT
5634                                                                  ;INHIBIT ECC CORRECTION
5635                                                                  ;DO NOT INHIBIT HEADER COMPARE
5636  020546  002350                        REFOR                   ;GET READY TO DO A REFOR
5637                                                                  ;READ HEADER AND DATA WITH 72 IN RHCS1
5638
5639
5640                                        ;*SAVE REGISTERS FOR COMPARISON AFTER ATTEMPTED READ
5641  020550  004037  033462                JSR     R0,@#SAVER      ;SAVE REGISTERS
5642  020554  002172                        RHWC                    ;RHWC IS THE FIRST REGISTER SAVED
5643  020556  004512                        SAVERE                  ;STARTING ADDRES OF WHERE
5644                                                                  ;THE REGISTERS ARE SAVED
5645  020560  000022                        18.                     ;NUMBER OF REGISTERS
5646                                                                  ;SAVED = 18.
5647
5648  020562  004737  033374                JSR     PC,@#CHECKT     ;CHECKS DVA,RDY,MOL,DPR,DRY AND VV = 1
5649                                                                  ;AND THAT ALL STATUS BITS ARE = 0
5650  020566  104401  057465                TYPE    ,CPHALT         ;CANNOT CONTINUE TESTING IF NOT
5651  020572  000000                        HALT                    ;STOP TEST
5652
5653  020574  013777  004506  161364        MOV     @#RP4VEC,@RPVEC ;SET RP04 VECTOR ADDRESS
5654                                                                  ;TO 'TIME1' IF P-CLOCK IS PRESENT
5655                                                                  ;OR TO 'TIME2' IF P-CLOCK IS NOT THERE
5656                                                                  ;'TIME' WILL ONLY SAVE
5657                                                                  ;CURRENT CYLINDER ADDRESS
5658                                                                  ;AND LOOK AHEAD REGISTERS
5659
5660  020602  013746  002350                MOV     @#REFOR,-(SP)   ;GET READY TO MOVE COMMAND
5661  020606  052716  000101                BIS     #GO!IE,(SP)     ;GET READY TO SET GO AND
```

```
                                                        ;ENABLE INTERRUPT
020612  012677  161362                MOV     (SP)+,@RHCS1    ;GO WITH
                                                          ;72 IN RHCS1 FOR READ DATA
                                                        ;WITH INTERRUPT ENABLED


020616  104413                        WAT                     ;WAIT FOR IAE BIT TO SET
020620  002202                        RHER1                   ;WAIT FOR RHER1 REGISTER
020622  002000                        IAE                     ;WAIT FOR IAE BIT IN RHER1 REGISTER
020624  000002                        2.                      ;ALLOW 20 MICRO SECONDS
020626  000002                        2.                      ;IAE MUST SET BETWEEN
                                                        ;00 AND 40 MICRO SECONDS

                                    ;*CHANGE SAVED REGISTERS TO EXPECTED VALUES



                                    ;*AS EXCEPTION IS ASSERTED BEFORE RUN IS LATCHED
                                    ;*RHWC,RHBA,RHCS1,RHCS2,  CANNOT BE PEREDETERMINED
                                    ;*THEY WILL VARY DEPENDING ON GATE DELAYS ON DIFFRENT UNITS

020630  017737  161336  004512        MOV     @RHWC,@#SAVERE          ;RHWC IS UNPREDICTABLE
                                                                ;AS EXPLAINED ABOVE
020636  017737  161332  004514        MOV     @RHBA,@#SAVERE+2        ;RHBA IS UNPREDICTABLE
                                                                ;AS EXPLAINED ABOVE
020644  017737  161326  004516        MOV     @RHCS2,@#SAVERE+4       ;RHCS2 IS UNPREDICTABLE
                                                                ;AS EXPLAINED ABOVE
020652  017737  161322  004520        MOV     @RHCS1,@#SAVERE+6       ;RHCS1 IS UNPREDICTABLE
                                                                ;AS EXPLAINED ABOVE

020660  004037  034204                JSR     R0,@#CHREG      ;CHANGE BITS IN SAVED REGISTER
020664  002222                        RHDS1                   ;CHANGE RHDS1 REGISTER

020666  000002                        2                       ;2 BIT/BITS TO BE CHANGED
020670  000001                        1                       ;NEW VALUE OF ATA IS 1
020672  100000                        ATA                     ;CHANGE ATA BIT
020674  000001                        1                       ;NEW VALUE OF ERR IS 1
020676  040000                        ERR                     ;CHANGE ERR BIT
020700  017737  161300  004524        MOV     @RHDST,@#SAVERE+12      ;RHDST IS INDETERMINATE SO IT IS NOT CHECKED

020706  004037  034204                JSR     R0,@#CHREG      ;CHANGE BITS IN SAVED REGISTER
020712  002202                        RHER1                   ;CHANGE RHER1 REGISTER

020714  000001                        1                       ;1 BIT/BITS TO BE CHANGED
020716  000001                        1                       ;NEW VALUE OF IAE IS 1
020720  002000                        IAE                     ;CHANGE IAE BIT

020722  053737  004644  004536        BIS     @#ATTENT,@#SAVERE+24 ;SET APPROPIATE 'ATA' BITS
                                                        ;FOR WORKING DRIVE IN
                                                        ;SAVED RHAS LOCATION

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

020730  004037  034312                JSR     R0,@#COMREG     ;COMPARE SAVED REGISTERS WITH
```

```
                                                        ;PRESENT VALUE
020734  004512                  SAVERE                  ;GOOD DATA SAVED IN 'SAVERE'
020736  002254                  WC                      ;TEST DATA STARTING FROM 'RHWC'
020740  000022                  18.                     ;18. REGISTERS TO BE COMPARED
020742  020746                  1$                      ;RETURN TO 1$ ON ERROR
020744  020752                  2$                      ;RETURN TO 2$ ON NO ERROR


020746  104054          1$:     ERROR   54              ;ATTEMPTED READ HEADER
020750  000207                  RTS     PC              ;AND DATA WITH INVALID
                                                        ;ADDRESS CAUSED IMPROPER
                                                        ;REGISTER CHANGE
                                                        ;GOOD DATA GIVES WHAT
                                                        ;SHOULD BE THERE
                                                        ;RECEIVED DATA GIVES
                                                        ;REGISTER CONTENTS
                                                        ;AFTER ATTEMPTED
                                                        ;READ
020752                  2$:
```

```
                                        ;;*****************************************************************
                                        ;*TEST 23        INVALID ADDRESS ERROR - RHER1  (BIT #10)IAE

                                        ;*      A READ DATA IS ATTEMPTED TO CYLINDER 0, TRACK 0
                                        ;*      SECTOR 20 - FORMAT 18 BITS PER WORD
                                        ;*      INVALID ADDRESS ERROR IAE BIT #10 IN RHER1
                                        ;*      SHOULD SET

                                        ;;*****************************************************************
020752  000004                  ST23:   SCOPE
020754  012706  001000                  MOV     #STACK,SP       ;RESET STACK
020760  012737  000023  004504          MOV     #23,@#TSTNM     ;SAVE TEST NUMBER

020766  004737  033314                  JSR     PC,@#CLDISK     ;SET R1-RHCS1, R2-RHCS2
                                                                ;R3-RHDS1, R4-RHER1
                                                                ;GIVE RH-11 INITIALIZE
                                                                ;SETUP UNIT NUMBER

                                        ;*FILL READ INTO BUFFER WITH 125252

020772  004037  033164                  JSR     R0,@#CLAREA     ;CLEAR 260 WORDS, FROM REINTO
020776  003434                          REINTO                  ;STARTING FROM REINTO
021000  000260                          260                     ;260 WORDS
021002  125252                          125252                  ;FILL WITH 125252

                                        ;*THE READ HEADER AND DATA COMMAND IS FILLED

021004  004037  035276                  JSR     R0,@#RUN        ;SETUP TO RUN FOR DATA COMMAND
021010  000000                          0                       ;CYLINDER 0
021012     024                          .BYTE   20.             ;SECTOR 20.
021013     000                          .BYTE   0               ;TRACK 0
021014  177400                          -256.                   ;WORD COUNT = 256.
021016  003434                          REINTO                  ;BUS ADDRESS
                                                                ;STARTING ADDRESS OF DATA
                                                                ;BUFFER = REINTO
021020  000000                          0                       ;DO NOT INHIBIT BUS ADDRESS INCREMENT
021022  004000                          ECI                     ;18 BITS PER WORD FORMAT
                                                                ;INHIBIT ECC CORRECTION
                                                                ;DO NOT INHIBIT HEADER COMPARE
021024  002346                          READAT                  ;GET READY TO DO A READAT
                                                        ;READ DATA WITH 70 IN RHCS1

                                        ;*SAVE REGISTERS FOR COMPARISON AFTER ATTEMPTED READ
021026  004037  033462                  JSR     R0,@#SAVER      ;SAVE REGISTERS
021032  002172                          RHWC                    ;RHWC IS THE FIRST REGISTER SAVED
021034  004512                          SAVERE                  ;STARTING ADDRES OF WHERE
                                                                ;THE REGISTERS ARE SAVED
021036  000022                          18.                     ;NUMBER OF REGISTERS
                                                                ;SAVED = 18.

021040  004737  033374                  JSR     PC,@#CHECKT     ;CHECKS DVA,RDY,MOL,DPR,DRY AND VV = 1
                                                                ;AND THAT ALL STATUS BITS ARE = 0
021044  104401  057465                  TYPE    ,CPHALT         ;CANNOT CONTINUE TESTING IF NOT
```

```
5794  021050  000000                          HALT                    ;STOP TEST
5795
5796  021052  013777  004506  161106          MOV     @#RP4VEC,@RPVEC ;SET RP04 VECTOR ADDRESS
5797                                                                  ;TO 'TIME1' IF P-CLOCK IS PRESENT
5798                                                                  ;OR TO 'TIME2' IF P-CLOCK IS NOT THERE
5799                                                                  ;'TIME' WILL ONLY SAVE
5800                                                                  ;CURRENT CYLINDER ADDRESS
5801                                                                  ;AND LOOK AHEAD REGISTERS
5802
5803
5804  021060  013746  002346                  MOV     @#READAT,-(SP)  ;GET READY TO MOVE COMMAND
5805  021064  052716  000101                  BIS     #GO!IE,(SP)     ;GET READY TO SET GO AND
5806                                                                  ;ENABLE INTERRUPT
5807  021070  012677  161104                  MOV     (SP)+,@RHCS1    ;GO WITH
5808                                                                   ;70 IN RHCS1 FOR READ DATA
5809                                                                  ;WITH INTERRUPT ENABLED
5810
5811
5812  021074  104413                          WAT                     ;WAIT FOR IAE BIT TO SET
5813  021076  002202                          RHER1                   ;WAIT FOR RHER1 REGISTER
5814  021100  002000                          IAE                     ;WAIT FOR IAE BIT IN RHER1 REGISTER
5815  021102  000002                          2.                      ;ALLOW 20 MICRO SECONDS
5816  021104  000002                          2.                      ;IAE MUST SET BETWEEN
5817                                                                  ;00 AND 40 MICRO SECONDS
5818
5819                                  ;*CHANGE SAVED REGISTERS TO EXPECTED VALUES
5820
5821  021106  004037  034204                  JSR     R0,@#CHREG      ;CHANGE BITS IN SAVED REGISTER
5822  021112  002200                          RHCS1                   ;CHANGE RHCS1 REGISTER
5823
5824  021114  000002                          2                       ;2 BIT/BITS TO BE CHANGED
5825  021116  000001                          1                       ;NEW VALUE OF SC IS 1
5826  021120  100000                          SC                      ;CHANGE SC BIT
5827  021122  000001                          1                       ;NEW VALUE OF TRE IS 1
5828  021124  040000                          TRE                     ;CHANGE TRE BIT
5829  021126  017737  161052  004524          MOV     @RHDST,@#SAVERE+12;RHDST IS UNPREDICTABLE
5830
5831  021134  004037  034204                  JSR     R0,@#CHREG      ;CHANGE BITS IN SAVED REGISTER
5832  021140  002222                          RHDS1                   ;CHANGE RHDS1 REGISTER
5833
5834  021142  000002                          2                       ;2 BIT/BITS TO BE CHANGED
5835  021144  000001                          1                       ;NEW VALUE OF ATA IS 1
5836  021146  100000                          ATA                     ;CHANGE ATA BIT
5837  021150  000001                          1                       ;NEW VALUE OF ERR IS 1
5838  021152  040000                          ERR                     ;CHANGE ERR BIT
5839
5840  021154  004037  034204                  JSR     R0,@#CHREG      ;CHANGE BITS IN SAVED REGISTER
5841  021160  002202                          RHER1                   ;CHANGE RHER1 REGISTER
5842
5843  021162  000001                          1                       ;1 BIT/BITS TO BE CHANGED
5844  021164  000001                          1                       ;NEW VALUE OF IAE IS 1
5845  021166  002000                          IAE                     ;CHANGE IAE BIT
5846
5847  021170  053737  004644  004536          BIS     @#ATTENT,@#SAVERE+24 ;SET APPROPIATE 'ATA' BITS
5848                                                                  ;FOR WORKING DRIVE IN
5849                                                                  ;SAVED RHAS LOCATION
```

```
5850
5851                                    ;*COMPARE REGISTERS BEFORE ATTEMPTED READ
5852                                    ;*DATA WITH AFTER ATTEMPTED READ DATA
5853
5854  021176  004037  034312            JSR     R0,@#COMREG     ;COMPARE SAVED REGISTERS WITH
5855                                                            ;PRESENT VALUE
5856  021202  004512                    SAVERE                  ;GOOD DATA SAVED IN 'SAVERE'
5857  021204  002254                    WC                      ;TEST DATA STARTING FROM 'RHWC'
5858  021206  000022                    18.                     ;18. REGISTERS TO BE COMPARED
5859  021210  021214                    1$                      ;RETURN TO 1$ ON ERROR
5860  021212  021220                    2$                      ;RETURN TO 2$ ON NO ERROR
5861
5862
5863  021214  104054          1$:       ERROR   54              ;ATTEMPTED READ
5864  021216  000207                    RTS     PC              ;DATA WITH INVALID
5865                                                            ;ADDRESS CAUSED IMPROPER
5866                                                            ;REGISTER CHANGE
5867                                                            ;GOOD DATA GIVES WHAT
5868                                                            ;SHOULD BE THERE
5869                                                            ;RECEIVED DATA GIVES
5870                                                            ;REGISTERS CONTENTS
5871                                                            ;AFTER ATTEMPTED
5872                                                            ;READ
5873  021220                    2$:
5874
```

```
                                        ;;*************************************************************************
                                        ;*TEST 24       ADDRESS OVERFLOW ERROR - RHER1 (BIT#9) AOE

                                        ;*      A WRITE HEADER AND DATA COMMAND IS GIVEN FOR CYLINDER 0,TRACK 0
                                        ;*      SECTOR 0, 256 WORDS OF 0
                                        ;*      NO CHECK IS DONE AFTER THIS WRITE
                                        ;*
                                        ;*      A WRITE HEADER AND DATA COMMAND IS GIVEN FOR
                                        ;*      CYLINDER 410./814., TRACK 18, SECTOR 21, 261 WORDS
                                        ;*
                                        ;*      ADDRESS OVERFLOW ERROR - RHER1 BIT#9 (AOE) SHOULD SET
                                        ;*      AFTER SECTOR 21 IS WRITTEN
                                        ;*      ALL REGISTERS ARE CHECKED
                                        ;*
                                        ;*      A READ HEADER AND DATA CYLINDER 410./814., TRACK 18, SECTOR 21,
                                        ;*      260+66+4=330  WORDS IS ISSUED
                                        ;*
                                        ;*      SECTOR 21 SHOULD BE READ CORRECTLY BUT NO MORE
                                        ;*      READS SHOULD HAPPEN, AGAIN THE 'AOE' BIT SHOULD SET
                                        ;*
                                        ;*      CYLINDER 0, TRACK 0, SECTOR 0 IS READ AND THERE
                                        ;*      SHOULD BE NO CHANGE IN DATA IN THIS SECTOR FROM
                                        ;*      THE LAST WRITE HEADER AND DATA COMMAND

                                        ;;*************************************************************************
021220  000004                        TST24:  SCOPE
021222  012706  001000                        MOV     #STACK,SP       ;RESET STACK
021226  012737  000024  004504                MOV     #24,@#TSTNM     ;SAVE TEST NUMBER

021234  004737  033314                        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

021240  004037  033140                        JSR     R0,@#FLHEAD     ;SAVE HEADER DATA IN WRFROM
021244  002370                        WRFROM                          ;LOCATION WHERE SAVED
021246  000004                        4                               ;NUMBER OF WORDS SAVED
021250  010000                        10000                           ;FIRST DATA WORD
021252  000000                        0                               ;SECOND DATA WORD
021254  000000                        0                               ;THIRD DATA WORD
021256  000000                        0                               ;FOURTH DATA WORD

                                        ;*FILL WRITE FROM BUFFER WITH DATA

021260  004037  033164                        JSR     R0,@#CLAREA     ;CLEAR 256. WORDS, FROM WRFROM+<4*2>
021264  002400                        WRFROM+<4*2>                            ;STARTING FROM WRFROM+<4*2>
021266  000400                        256.                            ;256. WORDS
021270  000000                        0                               ;FILL WITH 0


                                        ;*THE FIRST WRITE OPERATION IS DONE
                                        ;*FILL WRITE HEADER AND DATA COMMAND
```

```
                                ;;*****************************************************************
                                
021272  004037  035276                JSR     R0,@#RUN        ;SETUP TO RUN FOR DATA COMMAND
021276  000000                        0                       ;CYLINDER 0
021300     000                      .BYTE   0               ;SECTOR 0
021301     000                      .BYTE   0               ;TRACK 0
021302  177374                      -256.-4                 ;WORD COUNT (DATA) = 256. +
                                                            ;4 HEADER WORDS
021304  002370                      WRFROM                  ;BUS ADDRESS
                                                            ;STARTING ADDRESS OF DATA
                                                            ;BUFFER = WRFROM
021306  000000                      0                       ;DO NOT INHIBIT BUS ADDRESS INCREMENT
021310  010000                      FMT22                   ;16 BITS PER WORD FORMAT
                                                            ;DO NOT INHIBIT ECC CORRECTION
                                                            ;DO NOT INHIBIT HEADER COMPARE
021312  002344                      WRIFOR                  ;GET READY TO DO A WRIFOR
                                                    ;WRITE HEADER AND DATA WITH 62 IN RHCS1

021314  004737  033374                JSR     PC,@#CHECKT     ;CHECKS DVA,RDY,MOL,DPR,DRY AND VV = 1
                                                            ;AND THAT ALL STATUS BITS ARE = 0
021320  104401  057465                TYPE    ,CPHALT         ;CANNOT CONTINUE TESTING IF NOT
021324  000000                        HALT                    ;STOP TEST

021326  013777  004506  160632        MOV     @#RP4VEC,@RPVEC ;SET RP04 VECTOR ADDRESS
                                                            ;TO 'TIME1' IF P-CLOCK IS PRESENT
                                                            ;OR TO 'TIME2' IF P-CLOCK IS NOT THERE
                                                            ;'TIME' WILL ONLY SAVE
                                                            ;CURRENT CYLINDER ADDRESS
                                                            ;AND LOOK AHEAD REGISTERS

021334  013746  002344                MOV     @#WRIFOR,-(SP)  ;GET READY TO MOVE COMMAND
021340  052716  000101                BIS     #GO!IE,(SP)     ;GET READY TO SET GO AND
                                                            ;ENABLE INTERRUPT
021344  012677  160630                MOV     (SP)+,@RHCS1    ;GO WITH
                                                             ;62 IN RHCS1 FOR WRITE HEADER AND DATA
                                                            ;WITH INTERRUPT ENABLED

021350  104413                        WAT                     ;WAIT FOR RDY BIT TO SET
021352  002200                        RHCS1                   ;WAIT FOR RHCS1 REGISTER
021354  000200                        RDY                     ;WAIT FOR RDY BIT IN RHCS1 REGISTER
021356  004704                        2500.                   ;ALLOW 25000 MICRO SECONDS
021360  004704                        2500.                   ;RDY MUST SET BETWEEN
                                                            ;00 AND 50000 MICRO SECONDS

021362  004737  033314                JSR     PC,@#CLDISK     ;SET R1-RHCS1, R2-RHCS2
                                                            ;R3-RHDS1, R4-RHER1
                                                            ;GIVE RH-11 INITIALIZE
                                                            ;SETUP UNIT NUMBER


                                ;*CHECK THE DRIVE TYPE AND DO THE
                                ;*APPROPRIATE SECOND WRITE OPERATION

```

```
                                        ;*FILL WRITE FROM BUFFER WITH HEADER
                                ;;**************************************************************

                                ;;**************************************************************
021366  005737  004636                TST     @#RP06  ;TEST FOR RP06 DRIVE
021372  001411                        BEQ     15$             ;TREAT DRIVE AS AN RP04
                                                            ;TREAT AS AN RP06
                                ;;**************************************************************

021374  004037  033140                JSR     R0,@#FLHEAD     ;SAVE HEADER DATA IN WRFROM
021400  002370                        WRFROM                  ;LOCATION WHERE SAVED
021402  000004                        4                       ;NUMBER OF WORDS SAVED
021404  011456                        11456                   ;FIRST DATA WORD
021406  011025                        <18.*400>!<21.>                 ;SECOND DATA WORD
021410  000000                        0                       ;THIRD DATA WORD
021412  000000                        0                       ;FOURTH DATA WORD
021414  000410                        BR      16$             ;CONTINUE WITH THE SECOND WRITE

021416                          15$:

021416  004037  033140                JSR     R0,@#FLHEAD     ;SAVE HEADER DATA IN WRFROM
021422  002370                        WRFROM                  ;LOCATION WHERE SAVED
021424  000004                        4                       ;NUMBER OF WORDS SAVED
021426  010632                        10632                   ;FIRST DATA WORD
021430  011025                        <18.*400>!<21.>                 ;SECOND DATA WORD
021432  000000                        0                       ;THIRD DATA WORD
021434  000000                        0                       ;FOURTH DATA WORD
021436                          16$:                            ;CONTINUE WRITE

                                        ;*FILL WRITE FROM BUFFER WITH DATA - 65125

021436  004037  033164                JSR     R0,@#CLAREA     ;CLEAR 256. WORDS, FROM WRFROM+<4*2>
021442  002400                        WRFROM+<4*2>                    ;STARTING FROM WRFROM+<4*2>
021444  000400                        256.                    ;256. WORDS
021446  065125                        <26.*2000>!<18.*40>!<21.>                       ;FILL WITH <26.*2000>!<18.*40>!<


                                        ;*CHECK THE DRIVE TYPE AND
                                        ;*FILL WRITE FROM BUFFER WITH APPROPRIATE NEXT HEADER

                                        ;*THIS IS A NON EXISTANT HEADER AND SHOULD NOT BE WRITTEN
                                        ;*SINCE 'AOE' SHOULD INHIBIT THE WRITE OPERATION


                                ;;**************************************************************
021450  005737  004636                TST     @#RP06  ;TEST FOR RP06 DRIVE
021454  001411                        BEQ     17$             ;TREAT UNIT AS AN RP04
                                                            ;TREAT AS AN RP06
                                ;;**************************************************************


021456  004037  033140                JSR     R0,@#FLHEAD     ;SAVE HEADER DATA IN WRFROM+<260.*2>
```

```
6043  021462  003400                        WRFROM+<260.*2>                 ;LOCATION WHERE SAVED
6044  021464  000004                        4                               ;NUMBER OF WORDS SAVED
6045  021466  011457                        11457                           ;FIRST DATA WORD
6046  021470  000000                        0                               ;SECOND DATA WORD
6047  021472  000000                        0                               ;THIRD DATA WORD
6048  021474  000000                        0                               ;FOURTH DATA WORD
6049  021476  000410                        BR      18$                     ;CONTINUE WITH TEST
6050
6051
6052  021500                          17$:
6053
6054  021500  004037  033140                JSR     R0,@#FLHEAD             ;SAVE HEADER DATA IN WRFROM+<260.*2>
6055  021504  003400                        WRFROM+<260.*2>                 ;LOCATION WHERE SAVED
6056  021506  000004                        4                               ;NUMBER OF WORDS SAVED
6057  021510  010633                        10633                           ;FIRST DATA WORD
6058  021512  000000                        0                               ;SECOND DATA WORD
6059  021514  000000                        0                               ;THIRD DATA WORD
6060  021516  000000                        0                               ;FOURTH DATA WORD
6061  021520                          18$:                                  ;CONTINUE
6062                                        ;*FILL WRITE FROM BUFFER WITH DATA FOR NEXT SECTOR
6063
6064  021520  004037  033164                JSR     R0,@#CLAREA             ;CLEAR 2 WORDS, FROM WRFROM+<264.*2>
6065  021524  003410                        WRFROM+<264.*2>                 ;STARTING FROM WRFROM+<264.*2>
6066  021526  000002                        2                               ;2 WORDS
6067  021530  066000                        <27.*2000>                      ;FILL WITH <27.*2000>
6068
6069
6070
6071                                        ;*CHECK THE DRIVE TYPE AND DO THE APPROPRIATE
6072                                        ;*FILL WRITE HEADER AND DATA COMMAND
6073
6074
6075                                  ;;************************************************************************
6076  021532  005737  004636                TST     @#RP06  ;TEST FOR RP06 DRIVE
6077  021536  001412                        BEQ     7$                      ;TREAT UNIT AS AN RP04
6078                                                                        ;TREAT UNIT AS AN RP06
6079                                  ;;************************************************************************
6080
6081
6082  021540  004037  035276                JSR     R0,@#RUN                ;SETUP TO RUN FOR DATA COMMAND
6083  021544  001456                        814.                            ;CYLINDER 814.
6084  021546     025                        .BYTE   21.                     ;SECTOR 21.
6085  021547     022                        .BYTE   18.                     ;TRACK 18.
6086  021550  177373                        -257.-4                         ;WORD COUNT (DATA) = 257. +
6087                                                                        ;4 HEADER WORDS
6088  021552  002370                        WRFROM                          ;BUS ADDRESS
6089                                                                        ;STARTING ADDRESS OF DATA
6090                                                                        ;BUFFER = WRFROM
6091  021554  000000                        0                               ;DO NOT INHIBIT BUS ADDRESS INCREMENT
6092  021556  010000                        FMT22                           ;16 BITS PER WORD FORMAT
6093                                                                        ;DO NOT INHIBIT ECC CORRECTION
6094                                                                        ;DO NOT INHIBIT HEADER COMPARE
6095  021560  002344                        WRIFOR                          ;GET READY TO DO A WRIFOR
6096                                                              ;WRITE HEADER AND DATA WITH 62 IN RHCS1
6097
6098  021562  000411                        BR      8$
```

```
                        T24     ADDRESS OVERFLOW ERROR - RHER1 (BIT#9) AOE                                  SEQ 0129

 6099  021564                   7$:
 6100
 6101  021564  004037  035276           JSR     R0,@#RUN        ;SETUP TO RUN FOR DATA COMMAND
 6102  021570  000632                   410.                    ;CYLINDER 410.
 6103  021572     025                   .BYTE   21.             ;SECTOR 21.
 6104  021573     022                   .BYTE   18.             ;TRACK 18.
 6105  021574  177373                   -257.-4                 ;WORD COUNT (DATA) = 257. +
 6106                                                           ;4 HEADER WORDS
 6107  021576  002370                   WRFROM                  ;BUS ADDRESS
 6108                                                           ;STARTING ADDRESS OF DATA
 6109                                                           ;BUFFER = WRFROM
 6110  021600  000000                   0                       ;DO NOT INHIBIT BUS ADDRESS INCREMENT
 6111  021602  010000                   FMT22                   ;16 BITS PER WORD FORMAT
 6112                                                           ;DO NOT INHIBIT ECC CORRECTION
 6113                                                           ;DO NOT INHIBIT HEADER COMPARE
 6114  021604  002344                   WRIFOR                  ;GET READY TO DO A WRIFOR
 6115                                                  ;WRITE HEADER AND DATA WITH 62 IN RHCS1
 6116
 6117  021606                   8$:
 6118
 6119                                   ;*SAVE REGISTERS FOR COMPARISON AFTER WIRTE HEADER AND DATA
 6120  021606  004037  033462           JSR     R0,@#SAVER      ;SAVE REGISTERS
 6121  021612  002172                   RHWC                    ;RHWC IS THE FIRST REGISTER SAVED
 6122  021614  004512                   SAVERE                  ;STARTING ADDRES OF WHERE
 6123                                                           ;THE REGISTERS ARE SAVED
 6124  021616  000022                   18.                     ;NUMBER OF REGISTERS
 6125                                                           ;SAVED = 18.
 6126
 6127  021620  004737  033374           JSR     PC,@#CHECKT     ;CHECKS DVA,RDY,MOL,DPR,DRY AND VV = 1
 6128                                                           ;AND THAT ALL STATUS BITS ARE = 0
 6129  021624  104401  057465           TYPE    ,CPHALT         ;CANNOT CONTINUE TESTING IF NOT
 6130  021630  000000                   HALT                    ;STOP TEST
 6131
 6132  021632  013777  004506  160326   MOV     @#RP4VEC,@RPVEC ;SET RP04 VECTOR ADDRESS
 6133                                                           ;TO 'TIME1' IF P-CLOCK IS PRESENT
 6134                                                           ;OR TO 'TIME2' IF P-CLOCK IS NOT THERE
 6135                                                           ;'TIME' WILL ONLY SAVE
 6136                                                           ;CURRENT CYLINDER ADDRESS
 6137                                                           ;AND LOOK AHEAD REGISTERS
 6138
 6139
 6140  021640  013746  002344           MOV     @#WRIFOR,-(SP)  ;GET READY TO MOVE COMMAND
 6141  021644  052716  000101           BIS     #GO!IE,(SP)     ;GET READY TO SET GO AND
 6142                                                           ;ENABLE INTERRUPT
 6143  021650  012677  160324           MOV     (SP)+,@RHCS1    ;GO WITH
 6144                                                            ;62 IN RHCS1 FOR WRITE HEADER AND DATA
 6145                                                           ;WITH INTERRUPT ENABLED
 6146
 6147
 6148  021654  104413                   WAT                     ;WAIT FOR RDY BIT TO SET
 6149  021656  002200                   RHCS1                   ;WAIT FOR RHCS1 REGISTER
 6150  021660  000200                   RDY                     ;WAIT FOR RDY BIT IN RHCS1 REGISTER
 6151  021662  004704                   2500.                   ;ALLOW 25000 MICRO SECONDS
 6152  021664  004704                   2500.                   ;RDY MUST SET BETWEEN
 6153                                                           ;00 AND 50000 MICRO SECONDS
 6154
```

```
                                        ;*CHANGE SAVED REGISTERS TO EXPECTED VALUES

021666  004037  033216                  JSR     R0,@#FILLRE     ;MOV WRFROM+<260.*2>+<1.*2> INTO SAVED RHBA
021672  002174                          RHBA                    ;SAVED REGISTER TO CHANGE
021674  003402                          WRFROM+<260.*2>+<1.*2>                  ;DATA

021676  004037  033216                  JSR     R0,@#FILLRE     ;MOV 0 INTO SAVED RHWC
021702  002172                          RHWC                    ;SAVED REGISTER TO CHANGE
021704  000000                          0                       ;DATA

021706  004037  034204                  JSR     R0,@#CHREG      ;CHANGE BITS IN SAVED REGISTER
021712  002200                          RHCS1                   ;CHANGE RHCS1 REGISTER

021714  000002                          2                       ;2 BIT/BITS TO BE CHANGED
021716  000001                          1                       ;NEW VALUE OF SC IS 1
021720  100000                          SC                      ;CHANGE SC BIT
021722  000001                          1                       ;NEW VALUE OF TRE IS 1
021724  040000                          TRE                     ;CHANGE TRE BIT

021726  004037  034204                  JSR     R0,@#CHREG      ;CHANGE BITS IN SAVED REGISTER
021732  002176                          RHCS2                   ;CHANGE RHCS2 REGISTER

021734  000002                          2                       ;2 BIT/BITS TO BE CHANGED
021736  000001                          1                       ;NEW VALUE OF OR IS 1
021740  000200                          OR                      ;CHANGE OR BIT
021742  000001                          1                       ;NEW VALUE OF IR IS 1
021744  000100                          IR                      ;CHANGE IR BIT

021746  004037  033216                  JSR     R0,@#FILLRE     ;MOV AOE INTO SAVED RHER1
021752  002202                          RHER1                   ;SAVED REGISTER TO CHANGE
021754  001000                          AOE                     ;DATA

021756  004037  034204                  JSR     R0,@#CHREG      ;CHANGE BITS IN SAVED REGISTER
021762  002222                          RHDS1                   ;CHANGE RHDS1 REGISTER

021764  000003                          3                       ;3 BIT/BITS TO BE CHANGED
021766  000001                          1                       ;NEW VALUE OF ATA IS 1
021770  100000                          ATA                     ;CHANGE ATA BIT
021772  000001                          1                       ;NEW VALUE OF ERR IS 1
021774  040000                          ERR                     ;CHANGE ERR BIT
021776  000001                          1                       ;NEW VALUE OF LBT IS 1
022000  002000                          LBT                     ;CHANGE LBT BIT

022002  053737  004644  004536          BIS     @#ATTENT,@#SAVERE+24 ;SET APPROPIATE 'ATA' BITS
                                                                ;FOR WORKING DRIVE IN
                                                                ;SAVED RHAS LOCATION

                                        ;*CHECK DEVICE TYPE BEFORE SETTING UP 'RHCA' & 'RHCC'

                                ;;**************************************************************************
022010  005737  004636                  TST     @#RP06  ;TEST FOR RP06 DRIVE
022014  001411                          BEQ     9$              ;TREAT AS RP04
                                                                ;TREAT AS RP06
```

```
6211                                    ;;*****************************************************************
6212
6213
6214  022016  004037  033216                      JSR     R0,@#FILLRE     ;MOV 815. INTO SAVED RHCA
6215  022022  002212                              RHCA                    ;SAVED REGISTER TO CHANGE
6216  022024  001457                              815.                    ;DATA
6217
6218
6219  022026  004037  033216                      JSR     R0,@#FILLRE     ;MOV 814. INTO SAVED RHCC
6220  022032  002234                              RHCC                    ;SAVED REGISTER TO CHANGE
6221  022034  001456                              814.                    ;DATA
6222
6223  022036  000410                              BR      10$             ;CONTINUE WITH TEST
6224  022040                          9$:
6225
6226  022040  004037  033216                      JSR     R0,@#FILLRE     ;MOV 411. INTO SAVED RHCA
6227  022044  002212                              RHCA                    ;SAVED REGISTER TO CHANGE
6228  022046  000633                              411.                    ;DATA
6229
6230
6231  022050  004037  033216                      JSR     R0,@#FILLRE     ;MOV 410. INTO SAVED RHCC
6232  022054  002234                              RHCC                    ;SAVED REGISTER TO CHANGE
6233  022056  000632                              410.                    ;DATA
6234
6235  022060                          10$:                                ;CONTINUE WITH TEST
6236
6237
6238
6239  022060  017737  160120  004524              MOV     @RHDST,@#SAVERE+12 ;RHDST IS UNPREDICTABLE
6240
6241                                            ;*COMPARE REGISTERS BEFORE WRITE HEADER AND DATA WITH AFTER
6242
6243  022066  004037  034312                      JSR     R0,@#COMREG     ;COMPARE SAVED REGISTERS WITH
6244                                                                        ;PRESENT VALUE
6245  022072  004512                              SAVERE                  ;GOOD DATA SAVED IN 'SAVERE'
6246  022074  002254                              WC                      ;TEST DATA STARTING FROM 'RHWC'
6247  022076  000022                              18.                     ;18. REGISTERS TO BE COMPARED
6248  022100  022104                              1$                      ;RETURN TO 1$ ON ERROR
6249  022102  022110                              2$                      ;RETURN TO 2$ ON NO ERROR
6250
6251
6252  022104  104055                  1$:         ERROR   55              ;WRITING HEADER AND DATA WITH
6253  022106  000207                              RTS     PC              ;EXPECTED ADDRESS OVERFLOW ERROR
6254                                                                        ;CAUSED IMPROPER REGISTER
6255                                                                        ;CHANGE
6256                                                                        ;GOOD DATA GIVES WHAT SHOULD
6257                                                                        ;BE THERE
6258                                                                        ;RECEIVED DATA GIVES WHAT
6259                                                                        ;WAS THERE AFTER WRITE
6260                                                                        ;HEADER AND DATA
6261
```

```
                                        ;*NOW PREPARE TO DO A READ HEADER AND DATA
                                        ;*(THE FIRST READ OPERATION)
                                        ;*CHECK THE DRIVE TYPE AND FILL
                                        ;*WRITE FROM BUFFER WITH APPROPRIATE EXPECTED HEADER
                                ;;*****************************************************************
022110                          2$:
022110  004737  033314                  JSR     PC,@#CLDISK     ;SET R1-RHCS1, R2-RHCS2
                                                                ;R3-RHDS1, R4-RHER1
                                                                ;GIVE RH-11 INITIALIZE
                                                                ;SETUP UNIT NUMBER
                                ;;*****************************************************************
022114  005737  004636                  TST     @#RP06  ;TEST FOR RP06 DRIVE
022120  001411                          BEQ     19$             ;TREAT UNIT AS AN RP04
                                                                ;TREAT AS AN RP06
                                ;;*****************************************************************
022122  004037  033140                  JSR     R0,@#FLHEAD     ;SAVE HEADER DATA IN WRFROM
022126  002370                          WRFROM                  ;LOCATION WHERE SAVED
022130  000004                          4                       ;NUMBER OF WORDS SAVED
022132  011456                          11456                   ;FIRST DATA WORD
022134  011025                          <18.*400>!<21.>                 ;SECOND DATA WORD
022136  000000                          0                       ;THIRD DATA WORD
022140  000000                          0                       ;FOURTH DATA WORD
022142  000410                          BR      20$             ;CONTINUE WITH TEST

022144                          19$:
022144  004037  033140                  JSR     R0,@#FLHEAD     ;SAVE HEADER DATA IN WRFROM
022150  002370                          WRFROM                  ;LOCATION WHERE SAVED
022152  000004                          4                       ;NUMBER OF WORDS SAVED
022154  010632                          10632                   ;FIRST DATA WORD
022156  011025                          <18.*400>!<21.>                 ;SECOND DATA WORD
022160  000000                          0                       ;THIRD DATA WORD
022162  000000                          0                       ;FOURTH DATA WORD
022164                          20$:                            ;CONTINUE

                                        ;*FILL WRITE FROM BUFFER WITH EXPECTED DATA

022164  004037  033164                  JSR     R0,@#CLAREA     ;CLEAR 256. WORDS, FROM WRFROM+<4*2>
022170  002400                          WRFROM+<4*2>                    ;STARTING FROM WRFROM+<4*2>
022172  000400                          256.                    ;256. WORDS
022174  065125                          <26.*2000>!<18.*40>!<21.>                       ;FILL WITH <26.*2000>!<18.*40>!<

                                        ;*FILL WRITE FROM BUFFER WITH 377 FROM WORDS 261 TO 266

022176  004037  033164                  JSR     R0,@#CLAREA     ;CLEAR 6 WORDS, FROM WRFROM+<260.*2>
022202  003400                          WRFROM+<260.*2>                 ;STARTING FROM WRFROM+<260.*2>
```

```
6318  022204  000006                    6                       ;6 WORDS
6319  022206  000377                    377                     ;FILL WITH 377
6320
6321
6322                                    ;*CLEAR READ INTO BUFFER
6323
6324  022210  004037  033164            JSR     R0,@#CLAREA     ;CLEAR 266. WORDS, FROM REINTO
6325  022214  003434                    REINTO                  ;STARTING FROM REINTO
6326  022216  000412                    266.                    ;266. WORDS
6327  022220  000377                    377                     ;FILL WITH 377
6328
6329
6330
6331  022222  004737  033314            JSR     PC,@#CLDISK     ;SET R1-RHCS1, R2-RHCS2
6332                                                            ;R3-RHDS1, R4-RHER1
6333                                                            ;GIVE RH-11 INITIALIZE
6334                                                            ;SETUP UNIT NUMBER
6335
6336
6337                                    ;*CHECK THE DRIVE TYPE AND DO THE APPROPRIATE
6338                                    ;*FILL READ HEADER AND DATA COMMAND
6339
6340                                ;;**************************************************************
6341  022226  005737  004636            TST     @#RP06  ;TEST FOR RP06 DRIVE
6342  022232  001412                    BEQ     11$             ;TREAT UNIT AS AN RP04
6343                                                            ;TREAT UNIT AS AN RP06
6344                                ;;**************************************************************
6345
6346  022234  004037  035276            JSR     R0,@#RUN        ;SETUP TO RUN FOR DATA COMMAND
6347  022240  001456                    814.                    ;CYLINDER 814.
6348  022242     025                    .BYTE   21.             ;SECTOR 21.
6349  022243     022                    .BYTE   18.             ;TRACK 18.
6350  022244  177266                    -326.-4                 ;WORD COUNT (DATA) = 326. +
6351                                                            ;4 HEADER WORDS
6352  022246  003434                    REINTO                  ;BUS ADDRESS
6353                                                            ;STARTING ADDRESS OF DATA
6354                                                            ;BUFFER = REINTO
6355  022250  000000                    0                       ;DO NOT INHIBIT BUS ADDRESS INCREMENT
6356  022252  014000                    ECI!FMT22                       ;16 BITS PER WORD FORMAT
6357                                                            ;INHIBIT ECC CORRECTION
6358                                                            ;DO NOT INHIBIT HEADER COMPARE
6359  022254  002350                    REFOR                   ;GET READY TO DO A REFOR
6360                                                            ;READ HEADER AND DATA WITH 72 IN RHCS1
6361
6362  022256  000411                    BR      12$             ;CONTINUE
6363  022260                        11$:
6364
6365  022260  004037  035276            JSR     R0,@#RUN        ;SETUP TO RUN FOR DATA COMMAND
6366  022264  000632                    410.                    ;CYLINDER 410.
6367  022266     025                    .BYTE   21.             ;SECTOR 21.
6368  022267     022                    .BYTE   18.             ;TRACK 18.
6369  022270  177266                    -326.-4                 ;WORD COUNT (DATA) = 326. +
6370                                                            ;4 HEADER WORDS
6371  022272  003434                    REINTO                  ;BUS ADDRESS
6372                                                            ;STARTING ADDRESS OF DATA
6373                                                            ;BUFFER = REINTO
```

```
6374  022274  000000                          0                       ;DO NOT INHIBIT BUS ADDRESS INCREMENT
6375  022276  014000                          ECI!FMT22                       ;16 BITS PER WORD FORMAT
6376                                                                  ;INHIBIT ECC CORRECTION
6377                                                                  ;DO NOT INHIBIT HEADER COMPARE
6378  022300  002350                          REFOR                   ;GET READY TO DO A REFOR
6379                                                                  ;READ HEADER AND DATA WITH 72 IN RHCS1
6380
6381  022302                          12$:                            ;CONTINUE WITH TESTING
6382
6383
6384                                          ;*SAVE REGISTERS FOR COMPARISON AFTER
6385                                          ;*READ HEADER AND DATA
6386  022302  004037  033462                  JSR     R0,@#SAVER      ;SAVE REGISTERS
6387  022306  002172                          RHWC                    ;RHWC IS THE FIRST REGISTER SAVED
6388  022310  004512                          SAVERE                  ;STARTING ADDRES OF WHERE
6389                                                                  ;THE REGISTERS ARE SAVED
6390  022312  000022                          18.                     ;NUMBER OF REGISTERS
6391                                                                  ;SAVED = 18.
6392
6393  022314  004737  033374                  JSR     PC,@#CHECKT     ;CHECKS DVA,RDY,MOL,DPR,DRY AND VV = 1
6394                                                                  ;AND THAT ALL STATUS BITS ARE = 0
6395  022320  104401  057465                  TYPE    ,CPHALT         ;CANNOT CONTINUE TESTING IF NOT
6396  022324  000000                          HALT                    ;STOP TEST
6397
6398  022326  013777  004506  157632          MOV     @#RP4VEC,@RPVEC ;SET RP04 VECTOR ADDRESS
6399                                                                  ;TO 'TIME1' IF P-CLOCK IS PRESENT
6400                                                                  ;OR TO 'TIME2' IF P-CLOCK IS NOT THERE
6401                                                                  ;'TIME' WILL ONLY SAVE
6402                                                                  ;CURRENT CYLINDER ADDRESS
6403                                                                  ;AND LOOK AHEAD REGISTERS
6404
6405
6406  022334  013746  002350                  MOV     @#REFOR,-(SP)   ;GET READY TO MOVE COMMAND
6407  022340  052716  000101                  BIS     #GO!IE,(SP)     ;GET READY TO SET GO AND
6408                                                                  ;ENABLE INTERRUPT
6409  022344  012677  157630                  MOV     (SP)+,@RHCS1    ;GO WITH
6410                                                                    ;72 IN RHCS1 FOR READ DATA
6411                                                                  ;WITH INTERRUPT ENABLED
6412
6413
6414  022350  104413                          WAT                     ;WAIT FOR RDY BIT TO SET
6415  022352  002200                          RHCS1                   ;WAIT FOR RHCS1 REGISTER
6416  022354  000200                          RDY                     ;WAIT FOR RDY BiT IN RHCS1 REGISTER
6417  022356  001614                          908.                    ;ALLOW 9080 MICRO SECONDS
6418  022360  001507                          839.                    ;RDY MUST SET BETWEEN
6419                                                                  ;690 AND 17470 MICRO SECONDS
6420
6421                                          ;*CHANGE        SAVED REGISTERS TO EXPECTED VALUES
6422
6423  022362  004037  033216                  JSR     R0,@#FILLRE     ;MOV REINTO+<260.*2> INTO SAVED RHBA
6424  022366  002174                          RHBA                    ;SAVED REGISTER TO CHANGE
6425  022370  004444                          REINTO+<260.*2>                 ;DATA
6426
6427
6428  022372  004037  033216                  JSR     R0,@#FILLRE     ;MOV -70. INTO SAVED RHWC
6429  022376  002172                          RHWC                    ;SAVED REGISTER TO CHANGE
```

```
                                                -70.                    ;DATA

                        004037  034204          JSR     R0,@#CHREG      ;CHANGE BITS IN SAVED REGISTER
                        002200                  RHCS1                   ;CHANGE RHCS1 REGISTER

                        000002                  2                       ;2 BIT/BITS TO BE CHANGED
                        000001                  1                       ;NEW VALUE OF SC IS 1
                        100000                  SC                      ;CHANGE SC BIT
                        000001                  1                       ;NEW VALUE OF TRE IS 1
                        040000                  TRE                     ;CHANGE TRE BIT

                        004037  034204          JSR     R0,@#CHREG      ;CHANGE BITS IN SAVED REGISTER
                        002222                  RHDS1                   ;CHANGE RHDS1 REGISTER

                        000003                  3                       ;3 BIT/BITS TO BE CHANGED
                        000001                  1                       ;NEW VALUE OF ATA IS 1
                        100000                  ATA                     ;CHANGE ATA BIT
                        000001                  1                       ;NEW VALUE OF ERR IS 1
                        040000                  ERR                     ;CHANGE ERR BIT
                        000001                  1                       ;NEW VALUE OF LBT IS 1
                        002000                  LBT                     ;CHANGE LBT BIT

                        004037  033216          JSR     R0,@#FILLRE     ;MOV AOE INTO SAVED RHER1
                        002202                  RHER1                   ;SAVED REGISTER TO CHANGE
                        001000                  AOE                     ;DATA


                                                ;*CHECK DRIVE TYPE BEFORE SETTING UP 'RHCA'

                                ;;**************************************************************************
                        005737  004636          TST     @#RP06  ;TEST FOR RP06 DRIVE
                        001405                  BEQ     13$             ;TREAT UNIT AS AN RP04
                                                                        ;TREAT UNIT AS AN RP06
                                ;;**************************************************************************

                        004037  033216          JSR     R0,@#FILLRE     ;MOV 815. INTO SAVED RHCA
                        002212                  RHCA                    ;SAVED REGISTER TO CHANGE
                        001457                  815.                    ;DATA

                        000404                  BR      14$             ;CONTINUE
                                13$:
                        004037  033216          JSR     R0,@#FILLRE     ;MOV 411. INTO SAVED RHCA
                        002212                  RHCA                    ;SAVED REGISTER TO CHANGE
                        000633                  411.                    ;DATA

                                14$:                                    ;CONTINUE WITH TEST


                        053737  004644  004536  BIS     @#ATTENT,@#SAVERE+24 ;SET APPROPIATE 'ATA' BITS
                                                                        ;FOR WORKING DRIVE IN
                                                                        ;SAVED RHAS LOCATION
                        017737  157464  004524  MOV     @RHDST,@#SAVERE+12 ;RHDST IS UNPREDICTABLE
```

```
                                        ;*COMPARE REGISTERS BEFORE READ HEADER AND DATA WITH
                                        ;*REGISTERS AFTER COMMAND
                                
022522  004037  034312                  JSR     R0,@#COMREG     ;COMPARE SAVED REGISTERS WITH
                                                                ;PRESENT VALUE
022526  004512                          SAVERE                  ;GOOD DATA SAVED IN 'SAVERE'
022530  002254                          WC                      ;TEST DATA STARTING FROM 'RHWC'
022532  000022                          18.                     ;18. REGISTERS TO BE COMPARED
022534  022540                          3$                      ;RETURN TO 3$ ON ERROR
022536  022544                          4$                      ;RETURN TO 4$ ON NO ERROR
                                
022540  104055                  3$:     ERROR   55              ;READING HEADER AND DATA WITH
022542  000207                          RTS     PC              ;EXPECTED ADDRESS OVERFLOW
                                                                ;ERROR CAUSED IMPROPER
                                                                ;REGISTER CHANGE
                                                                ;GOOD DATA GIVES WHAT SHOULD
                                                                ;BE THERE
                                                                ;RECEIVED DATA GIVES WHAT
                                                                ;WAS THERE AFTER COMMAND
                                
                                        ;*NOW COMPARE THE DATA READ
022544                          4$:
                                
022544  004037  035342                  JSR     R0,@#COMPAR     ;COMPARE TWO BLOCKS OF MEMORY
022550  002370                          WRFROM                  ;GOOD DATA STARTS FROM WRFROM
022552  003434                          REINTO                  ;TEST DATA STARTS FROM REINTO
022554  000412                          266.                    ;266., WORDS TO BE COMPARED
022556  022562                          5$                      ;RETURN TO 5$ ON ERROR
022560  022566                          6$                      ;RETURN TO 6$ ON NO ERROR
                                
                                
022562  104056                  5$:     ERROR   56              ;DATA READ WITH AN EXPECTED
022564  000207                          RTS     PC              ;ADDRESS OVERFLOW ERROR
                                                                ;IS INCORRECT
                                                                ;WORD NO 1 TO 260 SHOULD
                                                                ;BE READ CORRECTLY
                                                                ;WORD NO 261 TO 266 SHOULD
                                                                ;NOT CHANGE DUE TO THE READ
                                
022566                          6$:
                                
022566  004737  033314                  JSR     PC,@#CLDISK     ;SET R1-RHCS1, R2-RHCS2
                                                                ;R3-RHDS1, R4-RHER1
                                                                ;GIVE RH-11 INITIALIZE
                                                                ;SETUP UNIT NUMBER
```

```
                                        ;*NOW PREPARE TO READ CYLINDER 0, SECTOR 0, TRACK 0
                                        ;*TO SEE THAT NOTHING GOT WRITTEN ON THERE
                                        ;*WITH THE ADDRESS OVER FLOW BIT SET (AOE)
                                    ;;*****************************************************************

                                        ;*FILL WRITE FROM BUFFER WITH EXPECTED HEADER

022572  004037  033140                  JSR     R0,@#FLHEAD     ;SAVE HEADER DATA IN WRFROM
022576  002370                          WRFROM                  ;LOCATION WHERE SAVED
022600  000004                          4                       ;NUMBER OF WORDS SAVED
022602  010000                          10000                   ;FIRST DATA WORD
022604  000000                          0                       ;SECOND DATA WORD
022606  000000                          0                       ;THIRD DATA WORD
022610  000000                          0                       ;FOURTH DATA WORD

022612  004037  033164                  JSR     R0,@#CLAREA     ;CLEAR 256. WORDS, FROM WRFROM+<4*2>
022616  002400                          WRFROM+<4*2>                    ;STARTING FROM WRFROM+<4*2>
022620  000400                          256.                    ;256. WORDS
022622  000000                          0                       ;FILL WITH 0

                                        ;*FILL READ INTO BUFFER WITH 377

022624  004037  033164                  JSR     R0,@#CLAREA     ;CLEAR 260. WORDS, FROM REINTO
022630  003434                          REINTO                  ;STARTING FROM REINTO
022632  000404                          260.                    ;260. WORDS
022634  000377                          377                     ;FILL WITH 377


022636  004737  033314                  JSR     PC,@#CLDISK     ;SET R1-RHCS1, R2-RHCS2
                                                                ;R3-RHDS1, R4-RHER1
                                                                ;GIVE RH-11 INITIALIZE
                                                                ;SETUP UNIT NUMBER

                                        ;*FILL COMMAND FOR READ HEADER AND DATA

022642  004037  035276                  JSR     R0,@#RUN        ;SETUP TO RUN FOR DATA COMMAND
022646  000000                          0                       ;CYLINDER 0
022650     000                          .BYTE   0               ;SECTOR 0
022651     000                          .BYTE   0               ;TRACK 0
022652  177374                          -256.-4                 ;WORD COUNT (DATA) = 256. +
                                                                ;4 HEADER WORDS
022654  003434                          REINTO                  ;BUS ADDRESS
                                                                ;STARTING ADDRESS OF DATA
                                                                ;BUFFER = REINTO
022656  000000                          0                       ;DO NOT INHIBIT BUS ADDRESS INCREMENT
022660  014000                          ECI!FMT22                       ;16 BITS PER WORD FORMAT
                                                                ;INHIBIT ECC CORRECTION
                                                                ;DO NOT INHIBIT HEADER COMPARE
022662  002350                          REFOR                   ;GET READY TO DO A REFOR
                                                                ;READ HEADER AND DATA WITH 72 IN RHCS1


                                        ;*SAVE REGISTERS FOR COMPARISON AFTER READ
```

```
6588  022664  004037  033462              JSR     R0,@#SAVER      ;SAVE REGISTERS
6589  022670  002172                      RHWC                    ;RHWC IS THE FIRST REGISTER SAVED
6590  022672  004512                      SAVERE                  ;STARTING ADDRES OF WHERE
6591                                                              ;THE REGISTERS ARE SAVED
6592  022674  000021                      17.                     ;NUMBER OF REGISTERS
6593                                                              ;SAVED = 17.
6594  022676  004737  033374              JSR     PC,@#CHECKT     ;CHECKS DVA,RDY,MOL,DPR,DRY AND VV = 1
6595                                                              ;AND THAT ALL STATUS BITS ARE = 0
6596  022702  104401  057465              TYPE    ,CPHALT         ;CANNOT CONTINUE TESTING IF NOT
6597  022706  000000                      HALT                    ;STOP TEST
6598
6599  022710  013777  004506  157250      MOV     @#RP4VEC,@RPVEC ;SET RP04 VECTOR ADDRESS
6600                                                              ;TO 'TIME1' IF P-CLOCK IS PRESENT
6601                                                              ;OR TO 'TIME2' IF P-CLOCK IS NOT THERE
6602                                                              ;'TIME' WILL ONLY SAVE
6603                                                              ;CURRENT CYLINDER ADDRESS
6604                                                              ;AND LOOK AHEAD REGISTERS
6605
6606
6607  022716  013746  002350              MOV     @#REFOR,-(SP)   ;GET READY TO MOVE COMMAND
6608  022722  052716  000101              BIS     #GO!IE,(SP)     ;GET READY TO SET GO AND
6609                                                              ;ENABLE INTERRUPT
6610  022726  012677  157246              MOV     (SP)+,@RHCS1    ;GO WITH
6611                                                                ;72 IN RHCS1 FOR READ DATA
6612                                                              ;WITH INTERRUPT ENABLED
6613
6614
6615  022732  104413                      WAT                     ;WAIT FOR RDY BIT TO SET
6616  022734  002200                      RHCS1                   ;WAIT FOR RHCS1 REGISTER
6617  022736  000200                      RDY                     ;WAIT FOR RDY BIT IN RHCS1 REGISTER
6618  022740  004704                      2500.                   ;ALLOW 25000 MICRO SECONDS
6619  022742  004704                      2500.                   ;RDY MUST SET BETWEEN
6620                                                              ;00 AND 50000 MICRO SECONDS
6621
6622                                      ;*CHANGE        REGISTERS TO EXPECTED VALUE
6623
6624  022744  004037  033216              JSR     R0,@#FILLRE     ;MOV 0 INTO SAVED RHWC
6625  022750  002172                      RHWC                    ;SAVED REGISTER TO CHANGE
6626  022752  000000                      0                       ;DATA
6627
6628
6629  022754  004037  033216              JSR     R0,@#FILLRE     ;MOV REINTO+<260.*2> INTO SAVED RHBA
6630  022760  002174                      RHBA                    ;SAVED REGISTER TO CHANGE
6631  022762  004444                      REINTO+<260.*2>                 ;DATA
6632
6633
6634  022764  004037  033216              JSR     R0,@#FILLRE     ;MOV 1 INTO SAVED RHDST
6635  022770  002204                      RHDST                   ;SAVED REGISTER TO CHANGE
6636  022772  000001                      1                       ;DATA
6637
6638
6639  022774                      ST22A:  ;COMPARE REGISTER BEFORE READ WITH AFTER
6640
6641  022774  004037  034312              JSR     R0,@#COMREG     ;COMPARE SAVED REGISTERS WITH
6642                                                              ;PRESENT VALUE
6643  023000  004512                      SAVERE                  ;GOOD DATA SAVED IN 'SAVERE'
```

```
6644  023002  002254                        WC                      ;TEST DATA STARTING FROM 'RHWC'
6645  023004  000021                        17.                     ;17. REGISTERS TO BE COMPARED
6646  023006  023012                        4$                      ;RETURN TO 4$ ON ERROR
6647  023010  023016                        1$                      ;RETURN TO 1$ ON NO ERROR
6648
6649
6650  023012  104031              4$:       ERROR   31              ;READ HEADER AND DATA ON
6651  023014  000207                        RTS     PC              ;CYLINDER 0, SECTOR 0
6652                                                                ;TRACK 0 AFTER A FORCED
6653                                                                ;ADDRESS OVER FLOW ERROR
6654                                                                ;CAUSED IMPROPER REGISTER
6655                                                                ;CHANGE
6656                                                                ;GOOD DATA GIVES WHAT
6657                                                                ;SHOULD BE THERE
6658                                                                ;RECEIVED DATA GIVES WHAT
6659                                                                ;WAS THERE AFTER READ
6660                                                                ;HEADER AND DATA
6661                                                                ;IF HEADER COMPARE ERROR
6662                                                                ;IS FOUND AND THE DATA
6663                                                                ;ERROR GIVES THE NEW
6664                                                                ;HEADER TO
6665                                                                ;CYLINDER 633/1457 (OCTAL)
6666                                                                ;THEN 'AOE' OVER FLOWED
6667                                                                ;INTO HERE
6668
6669                                        ;*COMPARE DATA/READ
6670  023016                        1$:
6671
6672  023016  004037  035342                JSR     R0,@#COMPAR     ;COMPARE TWO BLOCKS OF MEMORY
6673  023022  002370                        WRFROM                  ;GOOD DATA STARTS FROM WRFROM
6674  023024  003434                        REINTO                  ;TEST DATA STARTS FROM REINTO
6675  023026  000404                        260.                    ;260., WORDS TO BE COMPARED
6676  023030  023034                        2$                      ;RETURN TO 2$ ON ERROR
6677  023032  023040                        3$                      ;RETURN TO 3$ ON NO ERROR
6678
6679
6680  023034  104032              2$:       ERROR   32              ;READ HEADER AND DATA
6681  023036  000207                        RTS     PC              ;ON CYLINDER 0, TRACK 0
6682                                                                ;SECTOR 0 AFTER A FORCED
6683                                                                ;'AOE' ERROR CAUSED
6684                                                                ;AN ERROR
6685                                                                ;IF FIRST WORD IS
6686                                                                ;10633/11457 (OCTAL) THEN
6687                                                                ;'AOE' OVER FLOWED INTO HERE
6688  023040                        3$:
6689
6690
6691
```

```
                                ;;*****************************************************************
                                ;*TEST 25       FORMAT ERROR - RHER1 (BIT #4)FMT

                                ;*      AN ATTEMPT WILL BE MADE TO WRITE DATA ON CYLINDER 0
                                ;*      SECTOR 0 TRACK 0 WITH 18 BITS PER WORD WHEN THE
                                ;*      HEADER HAS 16 BITS PER WORD SET
                                ;*
                                ;*      THIS SHOULD GIVE FORMAT ERROR FER BIT #4 IN RHER1
                                ;*
                                ;*      THEN THIS SECTOR WILL BE READ IN THE CORRECT FORMAT
                                ;*      16 BITS PER WORD TO CHECK THAT NOTHING GOT WRITTEN
                                ;;*****************************************************************
023040  000004                TST25:  SCOPE
023042  012706  001000                MOV     #STACK,SP       ;RESET STACK
023046  012737  000025  004504        MOV     #25,@#TSTNM     ;SAVE TEST NUMBER

023054  004737  033314                JSR     PC,@#CLDISK     ;SET R1-RHCS1, R2-RHCS2
                                                              ;R3-RHDS1, R4-RHER1
                                                              ;GIVE RH-11 INITIALIZE
                                                              ;SETUP UNIT NUMBER

                                ;*FIRST WRITE HEADER AND DATA CYLINDER 0, TRACK 0, SECTOR 0
                                ;*FILL WRITE FROM BUFFER WITH HEADER

023060  004037  033140                JSR     R0,@#FLHEAD     ;SAVE HEADER DATA IN WRFROM
023064  002370                        WRFROM                  ;LOCATION WHERE SAVED
023066  000004                        4                       ;NUMBER OF WORDS SAVED
023070  010000                        10000                   ;FIRST DATA WORD
023072  000000                        0                       ;SECOND DATA WORD
023074  000000                        0                       ;THIRD DATA WORD
023076  000000                        0                       ;FOURTH DATA WORD

                                ;*FILL WRITE FROM BUFFER WITH DATA

023100  004037  033164                JSR     R0,@#CLAREA     ;CLEAR 266. WORDS, FROM WRFROM+<4*2>
023104  002400                        WRFROM+<4*2>                    ;STARTING FROM WRFROM+<4*2>
023106  000412                        266.                    ;266. WORDS
023110  000000                        0                       ;FILL WITH 0

                                ;*FILL COMMAND

023112  004037  035276                JSR     R0,@#RUN        ;SETUP TO RUN FOR DATA COMMAND
023116  000000                        0                       ;CYLINDER 0
023120     000                      .BYTE   0                 ;SECTOR 0
023121     000                      .BYTE   0                 ;TRACK 0
023122  177362                      -266.-4                   ;WORD COUNT (DATA) = 266. +
                                                              ;4 HEADER WORDS
023124  002370                      WRFROM                    ;BUS ADDRESS
                                                              ;STARTING ADDRESS OF DATA
                                                              ;BUFFER = WRFROM
023126  000000                      0                         ;DO NOT INHIBIT BUS ADDRESS INCREMENT
023130  010000                      FMT22                     ;16 BITS PER WORD FORMAT
                                                              ;DO NOT INHIBIT ECC CORRECTION
                                                              ;DO NOT INHIBIT HEADER COMPARE
```

```
6748  023132  002344                    WRIFOR                    ;GET READY TO DO A WRIFOR
6749                                                          ;WRITE HEADER AND DATA WITH 62 IN RHCS1
6750
6751
6752  023134  004737  033374            JSR     PC,@#CHECKT       ;CHECKS DVA,RDY,MOL,DPR,DRY AND VV = 1
6753                                                          ;AND THAT ALL STATUS BITS ARE = 0
6754  023140  104401  057465            TYPE    ,CPHALT           ;CANNOT CONTINUE TESTING IF NOT
6755  023144  000000                    HALT                      ;STOP TEST
6756
6757  023146  013777  004506  157012    MOV     @#RP4VEC,@RPVEC   ;SET RP04 VECTOR ADDRESS
6758                                                          ;TO 'TIME1' IF P-CLOCK IS PRESENT
6759                                                          ;OR TO 'TIME2' IF P-CLOCK IS NOT THERE
6760                                                          ;'TIME' WILL ONLY SAVE
6761                                                          ;CURRENT CYLINDER ADDRESS
6762                                                          ;AND LOOK AHEAD REGISTERS
6763
6764
6765  023154  013746  002344            MOV     @#WRIFOR,-(SP)    ;GET READY TO MOVE COMMAND
6766  023160  052716  000101            BIS     #GO!IE,(SP)       ;GET READY TO SET GO AND
6767                                                          ;ENABLE INTERRUPT
6768  023164  012677  157010            MOV     (SP)+,@RHCS1      ;GO WITH
6769                                                           ;62 IN RHCS1 FOR WRITE HEADER AND DATA
6770                                                          ;WITH INTERRUPT ENABLED
6771
6772
6773  023170  104413                    WAT                       ;WAIT FOR RDY BIT TO SET
6774  023172  002200                    RHCS1                     ;WAIT FOR RHCS1 REGISTER
6775  023174  000200                    RDY                       ;WAIT FOR RDY BIT IN RHCS1 REGISTER
6776  023176  004704                    2500.                     ;ALLOW 25000 MICRO SECONDS
6777  023200  004704                    2500.                     ;RDY MUST SET BETWEEN
6778                                                          ;00 AND 50000 MICRO SECONDS
6779
6780
6781  023202  004737  033314            JSR     PC,@#CLDISK       ;SET R1-RHCS1, R2-RHCS2
6782                                                          ;R3-RHDS1, R4-RHER1
6783                                                          ;GIVE RH-11 INITIALIZE
6784                                                          ;SETUP UNIT NUMBER
6785
```

```
 6786
 6787                                   ;*NOW PREPARE TO WRITE WITH WRONG FORMAT
 6788
 6789                                   ;*FILL WRITE FROM BUFFER
 6790
 6791  023206  004037  033164           JSR     R0,@#CLAREA     ;CLEAR 266. WORDS, FROM WRFROM
 6792  023212  002370                   WRFROM                  ;STARTING FROM WRFROM
 6793  023214  000412                   266.                    ;266. WORDS
 6794  023216  000377                   377                     ;FILL WITH 377
 6795
 6796
 6797                                   ;*FILL WRITE DATA COMMAND
 6798
 6799  023220  004037  035276           JSR     R0,@#RUN        ;SETUP TO RUN FOR DATA COMMAND
 6800  023224  000000                   0                       ;CYLINDER 0
 6801  023226     000                   .BYTE   0               ;SECTOR 0
 6802  023227     000                   .BYTE   0               ;TRACK 0
 6803  023230  177366                   -266.                   ;WORD COUNT = 266.
 6804  023232  002370                   WRFROM                  ;BUS ADDRESS
 6805                                                           ;STARTING ADDRESS OF DATA
 6806                                                           ;BUFFER = WRFROM
 6807  023234  000000                   0                       ;DO NOT INHIBIT BUS ADDRESS INCREMENT
 6808  023236  000000                   0                       ;18 BITS PER WORD FORMAT
 6809                                                           ;DO NOT INHIBIT ECC CORRECTION
 6810                                                           ;DO NOT INHIBIT HEADER COMPARE
 6811  023240  002342                   WRIDAT                  ;GET READY TO DO A WRIDAT
 6812                                                 ;WRITE DATA WITH 60 IN RHCS1
 6813
 6814
 6815                                   ;*SAVE REGISTERS FOR COMPARISON AFTER ATTEMPTED WRITE DATA
 6816                                   ;*WITH WRONG FORMAT
 6817  023242  004037  033462           JSR     R0,@#SAVER      ;SAVE REGISTERS
 6818  023246  002172                   RHWC                    ;RHWC IS THE FIRST REGISTER SAVED
 6819  023250  004512                   SAVERE                  ;STARTING ADDRES OF WHERE
 6820                                                           ;THE REGISTERS ARE SAVED
 6821  023252  000022                   18.                     ;NUMBER OF REGISTERS
 6822                                                           ;SAVED = 18.
 6823
 6824  023254  004737  033374           JSR     PC,@#CHECKT     ;CHECKS DVA,RDY,MOL,DPR,DRY AND VV = 1
 6825                                                           ;AND THAT ALL STATUS BITS ARE = 0
 6826  023260  104401  057465           TYPE    ,CPHALT         ;CANNOT CONTINUE TESTING IF NOT
 6827  023264  000000                   HALT                    ;STOP TEST
 6828
 6829  023266  013777  004506  156672   MOV     @#RP4VEC,@RPVEC ;SET RP04 VECTOR ADDRESS
 6830                                                           ;TO 'TIME1' IF P-CLOCK IS PRESENT
 6831                                                           ;OR TO 'TIME2' IF P-CLOCK IS NOT THERE
 6832                                                           ;'TIME' WILL ONLY SAVE
 6833                                                           ;CURRENT CYLINDER ADDRESS
 6834                                                           ;AND LOOK AHEAD REGISTERS
 6835
 6836
 6837  023274  013746  002342           MOV     @#WRIDAT,-(SP)  ;GET READY TO MOVE COMMAND
 6838  023300  052716  000101           BIS     #GO!IE,(SP)     ;GET READY TO SET GO AND
 6839                                                           ;ENABLE INTERRUPT
 6840  023304  012677  156670           MOV     (SP)+,@RHCS1    ;GO WITH
 6841                                                            ;60 IN RHCS1 FOR WRITE DATA
```

```
                                                        ;WITH INTERRUPT ENABLED


023310  104413                  WAT                   ;WAIT FOR RDY BIT TO SET
023312  002200                  RHCS1                 ;WAIT FOR RHCS1 REGISTER
023314  000200                  RDY                   ;WAIT FOR RDY BIT IN RHCS1 REGISTER
023316  001522                  850.                  ;ALLOW 8500 MICRO SECONDS
023320  001510                  840.                  ;RDY MUST SET BETWEEN
                                                      ;100 AND 16900 MICRO SECONDS

                                ;*CHANGE SAVED REGISTERS TO EXPECTED VALUE
023322  005737  004640          TST     @#RH70        ;RH70 CONTROLLER ?
023326  001411                  BEQ     JP3           ;IF NOT, SKIP NEXT RH70 CODE

023330  004037  033216          JSR     R0,@#FILLRE   ;MOV -258. INTO SAVED RHWC
023334  002172                  RHWC                  ;SAVED REGISTER TO CHANGE
023336  177376          VAR3:   -258.                 ;DATA

023340  004037  033216          JSR     R0,@#FILLRE   ;MOV WRFROM+<8.*2> INTO SAVED AREA
023344  002174                  RHBA                  ;SAVED REGISTER TO CHANGE
023346  002410          VAR4:   WRFROM+<8.*2>         ;DATA
023350  000410                  BR      JP4           ;SKIP NEXT RH11 CODE


023352                  JP3:

023352  004037  033216          JSR     R0,@#FILLRE   ;MOV -200. INTO SAVED RHWC
023356  002172                  RHWC                  ;SAVED REGISTER TO CHANGE
023360  177470                  -200.                 ;DATA


023362  004037  033216          JSR     R0,@#FILLRE   ;MOV WRFROM+<66.*2> INTO SAVED RHBA
023366  002174                  RHBA                  ;SAVED REGISTER TO CHANGE
023370  002574                  WRFROM+<66.*2>                ;DATA


023372                  JP4:

023372  004037  034204          JSR     R0,@#CHREG    ;CHANGE BITS IN SAVED REGISTER
023376  002200                  RHCS1                 ;CHANGE RHCS1 REGISTER

023400  000002                  2                     ;2 BIT/BITS TO BE CHANGED
023402  000001                  1                     ;NEW VALUE OF SC IS 1
023404  100000                  SC                    ;CHANGE SC BIT
023406  000001                  1                     ;NEW VALUE OF TRE IS 1
023410  040000                  TRE                   ;CHANGE TRE BIT

023412  004037  034204          JSR     R0,@#CHREG    ;CHANGE BITS IN SAVED REGISTER
023416  002176                  RHCS2                 ;CHANGE RHCS2 REGISTER

023420  000001                  1                     ;1 BIT/BITS TO BE CHANGED
023422  000001                  1                     ;NEW VALUE OF OR IS 1
023424  000200                  OR                    ;CHANGE OR BIT

023426  004037  034204          JSR     R0,@#CHREG    ;CHANGE BITS IN SAVED REGISTER
```

```
 6898  023432  002222                          RHDS1                   ;CHANGE RHDS1 REGISTER
 6899
 6900  023434  000002                          2                       ;2 BIT/BITS TO BE CHANGED
 6901  023436  000001                          1                       ;NEW VALUE OF ATA IS 1
 6902  023440  100000                          ATA                     ;CHANGE ATA BIT
 6903  023442  000001                          1                       ;NEW VALUE OF ERR IS 1
 6904  023444  040000                          ERR                     ;CHANGE ERR BIT
 6905
 6906  023446  004037  033216                  JSR     R0,@#FILLRE     ;MOV 1 INTO SAVED RHDST
 6907  023452  002204                          RHDST                   ;SAVED REGISTER TO CHANGE
 6908  023454  000001                          1                       ;DATA
 6909
 6910
 6911  023456  053737  004644  004536          BIS     @#ATTENT,@#SAVERE+24 ;SET APPROPIATE 'ATA' BITS
 6912                                                                  ;FOR WORKING DRIVE IN
 6913                                                                  ;SAVED RHAS LOCATION
 6914
 6915  023464  004037  034204                  JSR     R0,@#CHREG      ;CHANGE BITS IN SAVED REGISTER
 6916  023470  002202                          RHER1                   ;CHANGE RHER1 REGISTER
 6917
 6918  023472  000001                          1                       ;1 BIT/BITS TO BE CHANGED
 6919  023474  000001                          1                       ;NEW VALUE OF FER IS 1
 6920  023476  000020                          FER                     ;CHANGE FER BIT
 6921  023500  017746  156472                  MOV     @RHCS2,-(SP)    ;GET RHCS2
 6922  023504  042716  177477                  BIC     #^C<IR!OR>,(SP) ;KEEP IR AND OR
 6923  023510  042737  000300  004516          BIC     #IR!OR,@#SAVERE+4;CLEAR SAVED IR OR
 6924  023516  052637  004516                  BIS     (SP)+,@#SAVERE+4;SET OR IR AS REQUIRED
 6925
 6926
 6927
 6928                                          ;*COMPARE REGISTERS BEFORE WRITE DATA WITH AFTER ATTEMPT
 6929
 6930  023522  004037  034312                  JSR     R0,@#COMREG     ;COMPARE SAVED REGISTERS WITH
 6931                                                                  ;PRESENT VALUE
 6932  023526  004512                          SAVERE                  ;GOOD DATA SAVED IN 'SAVERE'
 6933  023530  002254                          WC                      ;TEST DATA STARTING FROM 'RHWC'
 6934  023532  000022                          18.                     ;18. REGISTERS TO BE COMPARED
 6935  023534  023540                          1$                      ;RETURN TO 1$ ON ERROR
 6936  023536  023544                          2$                      ;RETURN TO 2$ ON NO ERROR
 6937
 6938
 6939  023540  104057                  1$:     ERROR   57              ;ATTEMPTING TO WRITE DATA
 6940  023542  000207                          RTS     PC              ;WITH WRONG FORMAT BIT CAUSED
 6941                                                                  ;IMPROPER REGISTER CHANGE
 6942                                                                  ;GOOD DATA GIVES WHAT SHOULD
 6943                                                                  ;BE THERE
 6944                                                                  ;RECEIVED DATA GIVES WHAT WAS
 6945                                                                  ;THERE AFTER ATTEMPTED WRITE
 6946
 6947                                          ;*NOW PREPARE TO READ WITH CORRECT FORMAT TO CHECK
 6948                                          ;*THAT NOTHING GOT WRITTEN
 6949  023544                          2$:
 6950
 6951  023544  004737  033314                  JSR     PC,@#CLDISK     ;SET R1-RHCS1, R2-RHCS2
 6952                                                                  ;R3-RHDS1, R4-RHER1
 6953                                                                  ;GIVE RH-11 INITIALIZE
```

```
                                                        ;SETUP UNIT NUMBER
                                ;*FILL WRITE FROM BUFFER WITH EXPECTED DATA

023550  004037  033164              JSR     R0,@#CLAREA     ;CLEAR 256. WORDS, FROM WRFROM
023554  002370                      WRFROM                  ;STARTING FROM WRFROM
023556  000400                      256.                    ;256. WORDS
023560  000000                      0                       ;FILL WITH 0


                                ;*FILL READ INTO BUFFER WITH 125252

023562  004037  033164              JSR     R0,@#CLAREA     ;CLEAR 256. WORDS, FROM REINTO
023566  003434                      REINTO                  ;STARTING FROM REINTO
023570  000400                      256.                    ;256. WORDS
023572  125252                      125252                  ;FILL WITH 125252


                                ;*FILL COMMAND TO READ DATA

023574  004037  035276              JSR     R0,@#RUN        ;SETUP TO RUN FOR DATA COMMAND
023600  000000                      0                       ;CYLINDER 0
023602     000                      .BYTE   0               ;SECTOR 0
023603     000                      .BYTE   0               ;TRACK 0
023604  177400                      -256.                   ;WORD COUNT = 256.
023606  003434                      REINTO                  ;BUS ADDRESS
                                                            ;STARTING ADDRESS OF DATA
                                                            ;BUFFER = REINTO
023610  000000                      0                       ;DO NOT INHIBIT BUS ADDRESS INCREMENT
023612  014000                      FMT22!ECI                       ;16 BITS PER WORD FORMAT
                                                            ;INHIBIT ECC CORRECTION
                                                            ;DO NOT INHIBIT HEADER COMPARE
023614  002346                      READAT                  ;GET READY TO DO A READAT
                                                ;READ DATA WITH 70 IN RHCS1


                                ;*SAVE REGISTERS FOR COMPARISON AFTER NORMAL READ
023616  004037  033462              JSR     R0,@#SAVER      ;SAVE REGISTERS
023622  002172                      RHWC                    ;RHWC IS THE FIRST REGISTER SAVED
023624  004512                      SAVERE                  ;STARTING ADDRES OF WHERE
                                                            ;THE REGISTERS ARE SAVED
023626  000022                      18.                     ;NUMBER OF REGISTERS
                                                            ;SAVED = 18.

023630  004737  033374              JSR     PC,@#CHECKT     ;CHECKS DVA,RDY,MOL,DPR,DRY AND VV = 1
                                                            ;AND THAT ALL STATUS BITS ARE = 0
023634  104401  057465              TYPE    ,CPHALT         ;CANNOT CONTINUE TESTING IF NOT
023640  000000                      HALT                    ;STOP TEST

023642  013777  004506  156316      MOV     @#RP4VEC,@RPVEC ;SET RP04 VECTOR ADDRESS
                                                            ;TO 'TIME1' IF P-CLOCK IS PRESENT
                                                            ;OR TO 'TIME2' IF P-CLOCK IS NOT THERE
                                                            ;'TIME' WILL ONLY SAVE
                                                            ;CURRENT CYLINDER ADDRESS
                                                            ;AND LOOK AHEAD REGISTERS
```

```
 7010  023650  013746  002346                MOV     @#READAT,-(SP)  ;GET READY TO MOVE COMMAND
 7011  023654  052716  000101                BIS     #GO!IE,(SP)     ;GET READY TO SET GO AND
 7012                                                                  ;ENABLE INTERRUPT
 7013  023660  012677  156314                MOV     (SP)+,@RHCS1    ;GO WITH
 7014                                                                   ;70 IN RHCS1 FOR READ DATA
 7015                                                                  ;WITH INTERRUPT ENABLED
 7016
 7017
 7018  023664  104413                        WAT                     ;WAIT FOR RDY BIT TO SET
 7019  023666  002200                        RHCS1                   ;WAIT FOR RHCS1 REGISTER
 7020  023670  000200                        RDY                     ;WAIT FOR RDY BIT IN RHCS1 REGISTER
 7021  023672  001614                        908.                    ;ALLOW 9080 MICRO SECONDS
 7022  023674  001507                        839.                    ;RDY MUST SET BETWEEN
 7023                                                                  ;690 AND 17470 MICRO SECONDS
 7024
 7025                                        ;*CHANGE SAVED REGISTERS TO EXPECTED VALUE
 7026
 7027  023676  004037  033216                JSR     R0,@#FILLRE     ;MOV REINTO+<256.*2> INTO SAVED RHBA
 7028  023702  002174                        RHBA                    ;SAVED REGISTER TO CHANGE
 7029  023704  004434                        REINTO+<256.*2>                 ;DATA
 7030
 7031
 7032  023706  004037  033216                JSR     R0,@#FILLRE     ;MOV 0 INTO SAVED RHWC
 7033  023712  002172                        RHWC                    ;SAVED REGISTER TO CHANGE
 7034  023714  000000                        0                       ;DATA
 7035
 7036
 7037  023716  004037  033216                JSR     R0,@#FILLRE     ;MOV 1 INTO SAVED RHDST
 7038  023722  002204                        RHDST                   ;SAVED REGISTER TO CHANGE
 7039  023724  000001                        1                       ;DATA
 7040
 7041  023726  017746  156244                MOV     @RHCS2,-(SP)    ;GET RHCS2
 7042  023732  042716  177477                BIC     #^C<IR!OR>,(SP) ;KEEP IR AND OR
 7043  023736  042737  000300  004516        BIC     #IR!OR,@#SAVERE+4;CLEAR SAVED IR OR
 7044  023744  052637  004516                BIS     (SP)+,@#SAVERE+4;SET OR IR AS REQUIRED
 7045
 7046
 7047
 7048
 7049                                        ;*COMPARE REGISTERS BEFORE READ WITH AFTER
 7050
 7051  023750  004037  034312                JSR     R0,@#COMREG     ;COMPARE SAVED REGISTERS WITH
 7052                                                                  ;PRESENT VALUE
 7053  023754  004512                        SAVERE                  ;GOOD DATA SAVED IN 'SAVERE'
 7054  023756  002254                        WC                      ;TEST DATA STARTING FROM 'RHWC'
 7055  023760  000022                        18.                     ;18. REGISTERS TO BE COMPARED
 7056  023762  023766                        3$                      ;RETURN TO 3$ ON ERROR
 7057  023764  023772                        4$                      ;RETURN TO 4$ ON NO ERROR
 7058
 7059
 7060  023766  104033                3$:     ERROR   33              ;READ DATA AFTER AN
 7061  023770  000207                        RTS     PC              ;ATTEMPTED WRITE WITH WRONG
 7062                                                                  ;IMPROPER REGISTER CHANGE
 7063                                                                  ;FORMAT CAUSED
 7064                                                                  ;GOOD DATA GIVES WHAT SHOULD
 7065                                                                  ;BE THERE
```

```
7066                                                    ;RECEIVED DATA GIVES WHAT
7067                                                    ;WAS THERE AFTER READ
7068
7069                                    ;*COMPARE DATA READ AFTER ATTEMPTED WRITE WITH
7070                                    ;*WRONG FORMAT BIT
7071  023772                      4$:
7072
7073  023772  004037  035342            JSR     R0,@#COMPAR     ;COMPARE TWO BLOCKS OF MEMORY
7074  023776  002370                    WRFROM                  ;GOOD DATA STARTS FROM WRFROM
7075  024000  003434                    REINTO                  ;TEST DATA STARTS FROM REINTO
7076  024002  000400                    256.                    ;256., WORDS TO BE COMPARED
7077  024004  024010                    5$                      ;RETURN TO 5$ ON ERROR
7078  024006  024014                    6$                      ;RETURN TO 6$ ON NO ERROR
7079
7080
7081  024010  104034            5$:     ERROR   34              ;DATA READ AFTER AN ATTEMPT
7082  024012  000207                    RTS     PC              ;TO WRITE WITH WRONG FORMAT
7083                                                            ;WAS INCORRECT
7084
7085  024014                      6$:
7086
```

```
                                ;;*****************************************************************
                                ;*TEST 26       FORMAT ERROR - RHER1 (BIT #4)FMT

                                ;*      AN ATTEMPT IS MADE TO READ DATA WITH WRONG
                                ;*      FORMAT BIT
                                ;*
                                ;*      FORMAT ERROR BIT #4 IN RHER1 SHOULD SET
                                ;*      NO DATA SHOULD BE READ

                                ;;*****************************************************************
024014  000004                  TST26:  SCOPE
024016  012706  001000                  MOV     #STACK,SP       ;RESET STACK
024022  012737  000026  004504          MOV     #26,@#TSTNM     ;SAVE TEST NUMBER

024030  004737  033314                  JSR     PC,@#CLDISK     ;SET R1-RHCS1, R2-RHCS2
                                                                ;R3-RHDS1, R4-RHER1
                                                                ;GIVE RH-11 INITIALIZE
                                                                ;SETUP UNIT NUMBER

                                ;*FILL WRITE FROM BUFFER WITH 107070

024034  004037  033164                  JSR     R0,@#CLAREA     ;CLEAR 256. WORDS, FROM WRFROM
024040  002370                          WRFROM                  ;STARTING FROM WRFROM
024042  000400                          256.                    ;256. WORDS
024044  107070                          107070                  ;FILL WITH 107070

-

                                ;*FILL READ INTO BUFFER WITH 107070

024046  004037  033164                  JSR     R0,@#CLAREA     ;CLEAR 256. WORDS, FROM REINTO
024052  003434                          REINTO                  ;STARTING FROM REINTO
024054  000400                          256.                    ;256. WORDS
024056  107070                          107070                  ;FILL WITH 107070


                                ;*FILL COMMAND TO READ WITH WRONG FORMAT

024060  004037  035276                  JSR     R0,@#RUN        ;SETUP TO RUN FOR DATA COMMAND
024064  000000                          0                       ;CYLINDER 0
024066     000                          .BYTE   0               ;SECTOR 0
024067     000                          .BYTE   0               ;TRACK 0
024070  177400                          -256.                   ;WORD COUNT = 256.
024072  003434                          REINTO                  ;BUS ADDRESS
                                                                ;STARTING ADDRESS OF DATA
                                                                ;BUFFER = REINTO
024074  000000                          0                       ;DO NOT INHIBIT BUS ADDRESS INCREMENT
024076  004000                          ECI                     ;18 BITS PER WORD FORMAT
                                                                ;INHIBIT ECC CORRECTION
                                                                ;DO NOT INHIBIT HEADER COMPARE
024100  002346                          READAT                  ;GET READY TO DO A READAT
                                                        ;READ DATA WITH 70 IN RHCS1


                                ;*SAVE REGISTERS FOR COMPARAISON AFTER READ
024102  004037  033462                  JSR     R0,@#SAVER      ;SAVE REGISTERS
024106  002172                          RHWC                    ;RHWC IS THE FIRST REGISTER SAVED
```

```
7143  024110  004512                    SAVERE                  ;STARTING ADDRES OF WHERE
7144                                                            ;THE REGISTERS ARE SAVED
7145  024112  000022                    18.                     ;NUMBER OF REGISTERS
7146                                                            ;SAVED = 18.
7147
7148  024114  004737  033374            JSR     PC,@#CHECKT     ;CHECKS DVA,RDY,MOL,DPR,DRY AND VV = 1
7149                                                            ;AND THAT ALL STATUS BITS ARE = 0
7150  024120  104401  057465            TYPE    ,CPHALT         ;CANNOT CONTINUE TESTING IF NOT
7151  024124  000000                    HALT                    ;STOP TEST
7152
7153  024126  013777  004506  156032    MOV     @#RP4VEC,@RPVEC ;SET RP04 VECTOR ADDRESS
7154                                                            ;TO 'TIME1' IF P-CLOCK IS PRESENT
7155                                                            ;OR TO 'TIME2' IF P-CLOCK IS NOT THERE
7156                                                            ;'TIME' WILL ONLY SAVE
7157                                                            ;CURRENT CYLINDER ADDRESS
7158                                                            ;AND LOOK AHEAD REGISTERS
7159
7160
7161  024134  013746  002346            MOV     @#READAT,-(SP)  ;GET READY TO MOVE COMMAND
7162  024140  052716  000101            BIS     #GO!IE,(SP)     ;GET READY TO SET GO AND
7163                                                            ;ENABLE INTERRUPT
7164  024144  012677  156030            MOV     (SP)+,@RHCS1    ;GO WITH
7165                                                             ;70 IN RHCS1 FOR READ DATA
7166                                                            ;WITH INTERRUPT ENABLED
7167
7168
7169  024150  104413                    WAT                     ;WAIT FOR RDY BIT TO SET
7170  024152  002200                    RHCS1                   ;WAIT FOR RHCS1 REGISTER
7171  024154  000200                    RDY                     ;WAIT FOR RDY BIT IN RHCS1 REGISTER
7172  024156  001522                    850.                    ;ALLOW 8500 MICRO SECONDS
7173  024160  001510                    840.                    ;RDY MUST SET BETWEEN
7174                                                            ;100 AND 16900 MICRO SECONDS
7175
7176                                    ;*CHANGE SAVED REGISTERS TO EXPECTED VALUE
7177
7178
7179  024162  004037  034204            JSR     R0,@#CHREG      ;CHANGE BITS IN SAVED REGISTER
7180  024166  002200                    RHCS1                   ;CHANGE RHCS1 REGISTER
7181
7182  024170  000002                    2                       ;2 BIT/BITS TO BE CHANGED
7183  024172  000001                    1                       ;NEW VALUE OF SC IS 1
7184  024174  100000                    SC                      ;CHANGE SC BIT
7185  024176  000001                    1                       ;NEW VALUE OF TRE IS 1
7186  024200  040000                    TRE                     ;CHANGE TRE BIT
7187
7188  024202  004037  034204            JSR     R0,@#CHREG      ;CHANGE BITS IN SAVED REGISTER
7189  024206  002222                    RHDS1                   ;CHANGE RHDS1 REGISTER
7190
7191  024210  000002                    2                       ;2 BIT/BITS TO BE CHANGED
7192  024212  000001                    1                       ;NEW VALUE OF ATA IS 1
7193  024214  100000                    ATA                     ;CHANGE ATA BIT
7194  024216  000001                    1                       ;NEW VALUE OF ERR IS 1
7195  024220  040000                    ERR                     ;CHANGE ERR BIT
7196
7197  024222  004037  033216            JSR     R0,@#FILLRE     ;MOV 1 INTO SAVED RHDST
7198  024226  002204                    RHDST                   ;SAVED REGISTER TO CHANGE
```

```
7199  024230  000001                          1                       ;DATA
7200
7201
7202  024232  004037  034204                  JSR     R0,@#CHREG      ;CHANGE BITS IN SAVED REGISTER
7203  024236  002202                          RHER1                   ;CHANGE RHER1 REGISTER
7204
7205  024240  000001                          1                       ;1 BIT/BITS TO BE CHANGED
7206  024242  000001                          1                       ;NEW VALUE OF FER IS 1
7207  024244  000020                          FER                     ;CHANGE FER BIT
7208
7209  024246  053737  004644  004536          BIS     @#ATTENT,@#SAVERE+24 ;SET APPROPIATE 'ATA' BITS
7210                                                                  ;FOR WORKING DRIVE IN
7211                                                                  ;SAVED RHAS LOCATION
7212  024254  017746  155716                  MOV     @RHCS2,-(SP)    ;GET RHCS2
7213  024260  042716  177477                  BIC     #^C<IR!OR>,(SP) ;KEEP IR AND OR
7214  024264  042737  000300  004516          BIC     #IR!OR,@#SAVERE+4;CLEAR SAVED IR OR
7215  024272  052637  004516                  BIS     (SP)+,@#SAVERE+4;SET OR IR AS REQUIRED
7216
7217
7218
7219                                          ;*COMPARE REGISTERS BEFORE WRITE DATA WITH AFTER ATTEMPT
7220
7221  024276  004037  034312                  JSR     R0,@#COMREG     ;COMPARE SAVED REGISTERS WITH
7222                                                                  ;PRESENT VALUE
7223  024302  004512                          SAVERE                  ;GOOD DATA SAVED IN 'SAVERE'
7224  024304  002254                          WC                      ;TEST DATA STARTING FROM 'RHWC'
7225  024306  000022                          18.                     ;18. REGISTERS TO BE COMPARED
7226  024310  024314                          1$                      ;RETURN TO 1$ ON ERROR
7227  024312  024320                          2$                      ;RETURN TO 2$ ON NO ERROR
7228
7229
7230  024314  104057                  1$:     ERROR   57              ;ATTEMPTING TO READ DATA
7231  024316  000207                          RTS     PC              ;WITH WRONG FORMAT BIT CAUSED
7232                                                                  ;IMPROPER REGISTER CHANGE
7233                                                                  ;GOOD DATA GIVES WHAT SHOULD BE
7234                                                                  ;THERE
7235                                                                  ;RECEIVED DATA GIVES WHAT WAS THERE
7236                                                                  ;AFTER READ DATA
7237
7238                                          ;*COMPARE READ INTO BUFFER TO CHECK THAT NOTHING WAS READ
7239  024320                          2$:
7240
7241  024320  004037  035342                  JSR     R0,@#COMPAR     ;COMPARE TWO BLOCKS OF MEMORY
7242  024324  002370                          WRFROM                  ;GOOD DATA STARTS FROM WRFROM
7243  024326  003434                          REINTO                  ;TEST DATA STARTS FROM REINTO
7244  024330  000400                          256.                    ;256., WORDS TO BE COMPARED
7245  024332  024336                          3$                      ;RETURN TO 3$ ON ERROR
7246  024334  024342                          4$                      ;RETURN TO 4$ ON NO ERROR
7247
7248
7249  024336  104034                  3$:     ERROR   34              ;ATTEMPT TO READ
7250  024340  000207                          RTS     PC              ;WITH WRONG FORMAT BIT
7251                                                                  ;CHANGED READ INTO BUFFER
7252                                                                  ;GOOD DATA GIVES WHAT SHOULD
7253                                                                  ;BE THERE
7254                                                                  ;BAD DATA GIVES WHAT WAS
```

```
 7255                                                    ;THERE AFTER READ DATA
 7256
 7257  024342                          4$:
 7258
```

```
                                        ;;*****************************************************************
                                        ;*TEST 27       REGISTER MODIFICATION REFUSED - RHER1(BIT #2),RMR

                                        ;*      CYLINDER1 TRACK 0, SECTOR 0 WILL BE WRITTEN WITH
                                        ;*      200 WORDS OF 2000 BY A WRITE HEADER AND DATA COMMAND
                                        ;*
                                        ;*      THE HEADS WILL BE BROUGHT TO CYLINDER 0 BY A SEEK
                                        ;*
                                        ;*      A READ DATA COMMAND WILL BE GIVEN TO CYLINDER 1 TRACK 0
                                        ;*      SECTOR 0 150. WORDS. THIS WILL TAKE AT
                                        ;*      LEAST 7 MILI SECONDS.  IMMEDIATELY AFTER GO AT
                                        ;*      IMPLIED SEEK TIME, WRITE INTO A REGISTER WILL BE ATTMEPTED
                                        ;*      THEN READY WILL BE WAITED ON TO COMPLETE THE READ DATA
                                        ;*      THEN ALL REGISTERS WILL BE COMPARED AND THE DATA READ
                                        ;*      SHOULD BE GOOD
                                        ;*      THIS WILL BE REPEATED FOR RHCS1, RHER1, RHDST, RHER2
                                        ;*      RHOF, RHCA, RHER3

                                        ;;*****************************************************************
024342  000004                        TST27:  SCOPE
024344  012706  001000                        MOV     #STACK,SP       ;RESET STACK
024350  012737  000027  004504                MOV     #27,@#TSTNM     ;SAVE TEST NUMBER

024356  004737  033314                        JSR     PC,@#CLDISK     ;SET R1-RHCS1, R2-RHCS2
                                                                        ;R3-RHDS1, R4-RHER1
                                                                        ;GIVE RH-11 INITIALIZE
                                                                        ;SETUP UNIT NUMBER
024362  012737  002200  004652                MOV     #RHCS1,@#TMP0   ;FIRST REGISTER TO BE TESTED
024370  012737  000007  004660                MOV     #7,@#TMP5       ;NUMBER OF REGISTERS TO BE TESTED

                                        ;*PREPARE TO WRITE HEADER AND DATA CYLINDER 1, TRACK 0, SECTOR 0
                                        ;*FILL WRITE FROM BUFFER WITH HEADER

024376                                  ST22:

024376  004737  033314                        JSR     PC,@#CLDISK     ;SET R1-RHCS1, R2-RHCS2
                                                                        ;R3-RHDS1, R4-RHER1
                                                                        ;GIVE RH-11 INITIALIZE
                                                                        ;SETUP UNIT NUMBER

024402  004037  033140                        JSR     R0,@#FLHEAD     ;SAVE HEADER DATA IN WRFROM
024406  002370                        WRFROM                          ;LOCATION WHERE SAVED
024410  000004                        4                               ;NUMBER OF WORDS SAVED
024412  010001                        10001                           ;FIRST DATA WORD
024414  000000                        0                               ;SECOND DATA WORD
024416  000000                        0                               ;THIRD DATA WORD
024420  000000                        0                               ;FOURTH DATA WORD

                                        ;*FILL WRITE FROM BUFFER WITH DATA

024422  004037  033164                        JSR     R0,@#CLAREA     ;CLEAR 256. WORDS, FROM WRFROM+<4*2>
024426  002400                        WRFROM+<4*2>                            ;STARTING FROM WRFROM+<4*2>
024430  000400                        256.                            ;256. WORDS
024432  002000                        2000                            ;FILL WITH 2000
```

```
 7315                                          ;*FILL COMMAND
 7316
 7317  024434  004037  035276                 JSR     R0,@#RUN          ;SETUP TO RUN FOR DATA COMMAND
 7318  024440  000001                          1                         ;CYLINDER 1
 7319  024442     000                          .BYTE   0                 ;SECTOR 0
 7320  024443     000                          .BYTE   0                 ;TRACK 0
 7321  024444  177464                          -200.-4                   ;WORD COUNT (DATA) = 200. +
 7322                                                                    ;4 HEADER WORDS
 7323  024446  002370                          WRFROM                    ;BUS ADDRESS
 7324                                                                    ;STARTING ADDRESS OF DATA
 7325                                                                    ;BUFFER = WRFROM
 7326  024450  000000                          0                         ;DO NOT INHIBIT BUS ADDRESS INCREMENT
 7327  024452  010000                          FMT22                     ;16 BITS PER WORD FORMAT
 7328                                                                    ;DO NOT INHIBIT ECC CORRECTION
 7329                                                                    ;DO NOT INHIBIT HEADER COMPARE
 7330  024454  002344                          WRIFOR                    ;GET READY TO DO A WRIFOR
 7331                                                  ;WRITE HEADER AND DATA WITH 62 IN RHCS1
 7332
 7333
 7334  024456  004737  033374                 JSR     PC,@#CHECKT       ;CHECKS DVA,RDY,MOL,DPR,DRY AND VV = 1
 7335                                                                    ;AND THAT ALL STATUS BITS ARE = 0
 7336  024462  104401  057465                 TYPE    ,CPHALT           ;CANNOT CONTINUE TESTING IF NOT
 7337  024466  000000                          HALT                      ;STOP TEST
 7338
 7339  024470  013777  004506  155470         MOV     @#RP4VEC,@RPVEC   ;SET RP04 VECTOR ADDRESS
 7340                                                                    ;TO 'TIME1' IF P-CLOCK IS PRESENT
 7341                                                                    ;OR TO 'TIME2' IF P-CLOCK IS NOT THERE
 7342                                                                    ;'TIME' WILL ONLY SAVE
 7343                                                                    ;CURRENT CYLINDER ADDRESS
 7344                                                                    ;AND LOOK AHEAD REGISTERS
 7345
 7346
 7347  024476  013746  002344                 MOV     @#WRIFOR,-(SP)    ;GET READY TO MOVE COMMAND
 7348  024502  052716  000101                 BIS     #GO!IE,(SP)       ;GET READY TO SET GO AND
 7349                                                                    ;ENABLE INTERRUPT
 73    PC,@#CHECKT     ;CHECKS DVA,RDY,MOL,DPR,DRY AND VV = 1
 7366                                                                    ;AND THAT ALL STATUS BITS ARE = 0
 7367  024530  104401  057465                 TYPE    ,CPHALT           ;CANNOT CONTINUE TESTING IF NOT
 7368  024534  000000                          HALT                      ;STOP TEST
 7369
 7370
```

```
7371  024536  004037  033264            JSR     R0,@#SEEKCY     ;SEEK FOR
7372  024542  000000                    0                       ;CYLINDER 0
7373
7374
7375  024544  013777  004506  155414    MOV     @#RP4VEC,@RPVEC ;SET RP04 VECTOR ADDRESS
7376                                                            ;TO 'TIME1' IF P-CLOCK IS PRESENT
7377                                                            ;OR TO 'TIME2' IF P-CLOCK IS NOT THERE
7378                                                            ;'TIME' WILL ONLY SAVE
7379                                                            ;CURRENT CYLINDER ADDRESS
7380                                                            ;AND LOCK AHEAD REGISTERS
7381
7382
7383  024552  013746  002352            MOV     @#SEECOM,-(SP)  ;GET READY TO MOVE COMMAND
7384  024556  052716  000101            BIS     #GO!IE,(SP)     ;GET READY TO SET GO AND
7385                                                            ;ENABLE INTERRUPT
7386  024562  012677  155412            MOV     (SP)+,@RHCS1    ;GO WITH
7387                                                             ;4 IN RHCS1 FOR SEEK
7388                                                            ;WITH INTERRUPT ENABLED
7389
7390
7391  024566  104413                    WAT                     ;WAIT FOR DRY BIT TO SET
7392  024570  002222                    RHDS1                   ;WAIT FOR RHDS1 REGISTER
7393  024572  000200                    DRY                     ;WAIT FOR DRY BIT IN RHDS1 REGISTER
7394  024574  002776                    1534.                   ;ALLOW 15340 MICRO SECONDS
7395  024576  001502                    834.                    ;DRY MUST SET BETWEEN
7396                                                            ;7000 AND 23680 MICRO SECONDS
7397
7398                                    ;*PREPARE FOR A READ DATA
7399
7400                                    ;*FILL WRITE FROM BUFFER WITH EXPECTED DATA FROM READ
7401
7402  024600  004037  033164            JSR     R0,@#CLAREA     ;CLEAR 150. WORDS, FROM WRFROM
7403  024604  002370                    WRFROM                  ;STARTING FROM WRFROM
7404  024606  000226                    150.                    ;150. WORDS
7405  024610  002000                    2000                    ;FILL WITH 2000
7406
7407
7408  024612  004037  033164            JSR     R0,@#CLAREA     ;CLEAR 106. WORDS, FROM WRFROM+<150.*2>
7409  024616  003044                    WRFROM+<150.*2>                 ;STARTING FROM WRFROM+<150.*2>
7410  024620  000152                    106.                    ;106. WORDS
7411  024622  000077                    77                      ;FILL WITH 77
7412
7413
7414                                    ;*FILL READ INTO BUFFER WITH DATA OTHER THAN WHAT IS EXPECTED
7415
7416  024624  004037  033164            JSR     R0,@#CLAREA     ;CLEAR 256. WORDS, FROM REINTO
7417  024630  003434                    REINTO                  ;STARTING FROM REINTO
7418  024632  000400                    256.                    ;256. WORDS
7419  024634  000077                    77                      ;FILL WITH 77
7420
7421                                    ;*FILL READ DATA COMMAND
7422
7423  024636  004037  035276            JSR     R0,@#RUN        ;SETUP TO RUN FOR DATA COMMAND
7424  024642  000001                    1                       ;CYLINDER 1
7425  024644     000                    .BYTE   0               ;SECTOR 0
7426  024645     000                    .BYTE   0               ;TRACK 0
```

```
7427  024646  177552                    -150.                   ;WORD COUNT = 150.
7428  024650  003434                    REINTO                  ;BUS ADDRESS
7429                                                            ;STARTING ADDRESS OF DATA
7430                                                            ;BUFFER = REINTO
7431  024652  000000                    0                       ;DO NOT INHIBIT BUS ADDRESS INCREMENT
7432  024654  014000                    ECI!FMT22                       ;16 BITS PER WORD FORMAT
7433                                                            ;INHIBIT ECC CORRECTION
7434                                                            ;DO NOT INHIBIT HEADER COMPARE
7435  024656  002346                    READAT                  ;GET READY TO DO A READAT
7436                                                  ;READ DATA WITH 70 IN RHCS1
7437
7438
7439                                    ;*SAVE REGISTERS FOR COMPARISON AFTER ATTEMPTED WRITE
7440                                    ;*INTO A REGISTER WHILE THE READ IS GOING ON
7441  024660  004037  033462            JSR     R0,@#SAVER      ;SAVE REGISTERS
7442  024664  002172                    RHWC                    ;RHWC IS THE FIRST REGISTER SAVED
7443  024666  004512                    SAVERE                  ;STARTING ADDRES OF WHERE
7444                                                            ;THE REGISTERS ARE SAVED
7445  024670  000022                    17459
7460
7461  024712  013746  002346            MOV     @#READAT,-(SP)  ;GET READY TO MOVE COMMAND
7462  024716  052716  000101            BIS     #GO!IE,(SP)     ;GET READY TO SET GO AND
7463                                                            ;ENABLE INTERRUPT
7464  024722  012677  155252            MOV     (SP)+,@RHCS1    ;GO WITH
7465                                                             ;70 IN RHCS1 FOR READ DATA
7466                                                            ;WITH INTERRUPT ENABLED
7467
7468  024726  013700  004652            MOV     @#TMP0,R0               ;SET UP R0 FOR WRITE
7469  024732  012730  002006            MOV     #BIT1!BIT2!BIT10,@(R0)+ ;ATTEMPT TO WRITE INTO
7470                                                                    ;REGISTERS DURING IMPLIED SEEK
7471  024736  010037  004652            MOV     R0,@#TMP0               ;SAVE OFF  R0
7472
7473                                    ;*NOW RMR IS SET BUT THE COMPLETION OF READ MUST BE
7474                                    ;*WAITED ON
7475
7476
7477  024742  104413                    WAT                     ;WAIT FOR RDY BIT TO SET
7478  024744  002200                    RHCS1                   ;WAIT FOR RHCS1 REGISTER
7479  020746  000200                    RDY                     ;WAIT FOR RDY BIT IN RHCS1 REGISTER
7480  024750  002237                    1183.                   ;ALLOW 11830 MICRO SECONDS
7481  024752  002237                    1183.                   ;RDY MUST SET BETWEEN
7482                                                            ;00 AND 23660 MICRO SECONDS
```

```
7483
7484                                    ;*CHANGE SAVED REGISTERS TO EXPECTED VALUE
7485
7486  024754  004037  034204            JSR     R0,@#CHREG      ;CHANGE BITS IN SAVED REGISTER
7487  024760  002234                    RHCC                    ;CHANGE RHCC REGISTER
7488
7489  024762  000001                    1                       ;1 BIT/BITS TO BE CHANGED
7490  024764  000001                    1                       ;NEW VALUE OF BIT0 IS 1
7491  024766  000001                    BIT0                    ;CHANGE BIT0 BIT
7492
7493  024770  004037  034204            JSR     R0,@#CHREG      ;CHANGE BITS IN SAVED REGISTER
7494  024774  002200                    RHCS1                   ;CHANGE RHCS1 REGISTER
7495
7496  024776  000001                    1                       ;1 BIT/BITS TO BE CHANGED
7497  025000  000001                    1                       ;NEW VALUE OF SC IS 1
7498  025002  100000                    SC                      ;CHANGE SC BIT
7499
7500  025004  004037  033216            JSR     R0,@#FILLRE     ;MOV 1 INTO SAVED RHDST
7501  025010  002204                    RHDST                   ;SAVED REGISTER TO CHANGE
7502  025012  000001                    1                       ;DATA
7503
7504
7505  025014  004037  034204            JSR     R0,@#CHREG      ;CHANGE BITS IN SAVED REGISTER
7506  025020  002222                    RHDS1                   ;CHANGE RHDS1 REGISTER
7507
7508  025022  000002                    2                       ;2 BIT/BITS TO BE CHANGED
7509  025024  000001                    1                       ;NEW VALUE OF ATA IS 1
7510  025026  100000                    ATA                     ;CHANGE ATA BIT
7511  025030  000001                    1                       ;NEW VALUE OF ERR IS 1
7512  025032  040000                    ERR                     ;CHANGE ERR BIT
7513
7514  025034  053737  004644  004536    BIS     @#ATTENT,@#SAVERE+24 ;SET APPROPIATE 'ATA' BITS
7515                                                            ;FOR WORKING DRIVE IN
7516                                                            ;SAVED RHAS LOCATION
7517
7518  025042  004037  034204            JSR     R0,@#CHREG      ;CHANGE BITS IN SAVED REGISTER
7519  025046  002202                    RHER1                   ;CHANGE RHER1 REGISTER
7520
7521  025050  000001                    1                       ;1 BIT/BITS TO BE CHANGED
7522  025052  000001                    1                       ;NEW VALUE OF RMR IS 1
7523  025054  000004                    RMR                     ;CHANGE RMR BIT
7524
7525  025056  004037  033216            JSR     R0,@#FILLRE     ;MOV REINTO+<150.*2> INTO SAVED RHBA
7526  025062  002174                    RHBA                    ;SAVED REGISTER TO CHANGE
7527  025064  004110                    REINTO+<150.*2>                 ;DATA
7528
7529
7530  025066  004037  033216            JSR     R0,@#FILLRE     ;MOV 0 INTO SAVED RHWC
7531  025072  002172                    RHWC                    ;SAVED REGISTER TO CHANGE
7532  025074  000000                    0                       ;DATA
7533
7534
7535                                    ;*COMPARE REGISTERS BEFORE READ DATA WITH REGISTERS
7536                                    ;*AFTER READ AND ATTEMPTED MODIFICATION OF REGISTER
7537
7538  025076  004037  034312            JSR     R0,@#COMREG     ;COMPARE SAVED REGISTERS WITH
```

```
                                                        ;PRESENT VALUE
025102  004512                  SAVERE                  ;GOOD DATA SAVED IN 'SAVERE'
025104  002254                  WC                      ;TEST DATA STARTING FROM 'RHWC'
025106  000022                  18.                     ;18. REGISTERS TO BE COMPARED
025110  025114                  2$                      ;RETURN TO 2$ ON ERROR
025112  025134                  3$                      ;RETURN TO 3$ ON NO ERROR

025114                  2$:
025114  010046                  MOV     R0,-(SP)        ;;PUSH R0 ON STACK
025116  013700  004652          MOV     @#TMP0,R0       ;GET REGISTER BEEING MODIFYED + 2 POINTER
025122  014037  001122          MOV     -(R0),@#$BDADR  ;GET ADDRESS OF REGISTER BEING MODIFYED
025126  104060                  ERROR   60              ;ATTEMPTING TO MODIFY REGISTER
025130  012600                  MOV     (SP)+,R0        ;;POP STACK INTO R0
025132  000207                  RTS     PC              ;DURING A READ COMMAND CAUSED
                                                        ;IMPROPER REGISTER CHANGE
                                                        ;GOOD DATA GIVES WHAT SHOULD
                                                        ;BE THERE
                                                        ;RECEIVED DATA GIVES WHAT WAS
                                                        ;THERE AFTER READ
                                ;*COMPARE DATA READ
025134                  3$:
025134  004037  035342          JSR     R0,@#COMPAR     ;COMPARE TWO BLOCKS OF MEMORY
025140  002370                  WRFROM                  ;GOOD DATA STARTS FROM WRFROM
025142  003434                  REINTO                  ;TEST DATA STARTS FROM REINTO
025144  025150                  4$                      ;4$, WORDS TO BE COMPARED
025146  025154                  ST23                    ;RETURN TO ST23 ON ERROR
                                                        ;RETURN TO  ON NO ERROR


025150  104034          4$:     ERROR   34              ;DATA READ WITH AN ATTEMPTED
025152  000207                  RTS     PC              ;MODIFICATION OF REGISTER
                                                        ;DURING READ CAUSED ERROR
025154  005337  004660  ST23:   DEC     @#TMP5          ;COUNT DOWN
025160  001002                  BNE     1$              ;BRANCH IF 7 NOT DONE
025162  000137  025172          JMP     TST30           ;JUMP TO NEXT TEST
025166  000137  024376  1$:     JMP     @#ST22          ;JUMP TO BEGINING OF TEST
```

```
                                ;;*****************************************************************
                                ;*TEST 30       REGISTER MODIFICATION REFUSED - RHER1 (BIT #2),'RMR'

                                ;*      A WRITE HEADER AND DATA COMMAND WILL BE GIVEN TO
                                ;*      CYLINDER 1 SECTOR 0 TRACK 0 DATA WORDS
                                ;*      OF 070707
                                ;*
                                ;*      A WRITE DATA COMMAND WILL BE GIVEN TO CYLINDER 1

                                ;*      SECTOR 0, TRACK 0, 256 WORDS OF 2000
                                ;*      AND 4 WORDS OF 2001. IMMEDIATELY AFTER GO
                                ;*      AN ATTEMPT WILL BE MADE TO MODIFY A REGISTER
                                ;*      RMR BIT #2 IN RHER1 SHOULD SET
                                ;*
                                ;*      AFTER THE WRITE IS COMPLETE ALL REGISTERS WILL
                                ;*      BE CHECKED
                                ;*
                                ;*      THE DATA WRITTEN WILL BE READ BACK AND CHECKED
                                ;*
                                ;*      THIS WILL BE REPEATED FOR RHCS1, RHER1, RHDST,
                                ;*      RHER2, RHOF, RHCA, RHER3

                                ;;*****************************************************************
025172  000004                  TST30:  SCOPE
025174  012706  001000                  MOV     #STACK,SP       ;RESET STACK
025200  012737  000030  004504          MOV     #30,@#TSTNM     ;SAVE TEST NUMBER

025206  004737  033314                  JSR     PC,@#CLDISK     ;SET R1-RHCS1, R2-RHCS2
                                                                ;R3-RHDS1, R4-RHER1
                                                                ;GIVE RH-11 INITIALIZE
                                                                ;SETUP UNIT NUMBER

025212  012737  002200  004652          MOV     #RHCS1,@#TMP0   ;FILL REGISTER TO BE MODIFIED
025220  012737  000007  004660          MOV     #7,@#TMP5               ;NUMBER OF REGISTERS TO BE TESTED

                                ;*PREPARE TO WRITE HEADER AND DATA

025226                          ST24:

025226  004737  033314                  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

025232  004037  033140                  JSR     R0,@#FLHEAD     ;SAVE HEADER DATA IN WRFROM
025236  002370                  WRFROM                          ;LOCATION WHERE SAVED
025240  000004                  4                               ;NUMBER OF WORDS SAVED
025242  010001                  10001                           ;FIRST DATA WORD
025244  000000                  0                               ;SECOND DATA WORD
025246  000000                  0                               ;THIRD DATA WORD
025250  000000                  0                               ;FOURTH DATA WORD

                                ;*FILL WRITE FROM BUFFER WITH DATA
```

```
7634
7635  025252  004037  033164          JSR     R0,@#CLAREA     ;CLEAR 256. WORDS, FROM WRFROM+<4*2>
7636  025256  002400                  WRFROM+<4*2>                    ;STARTING FROM WRFROM+<4*2>
7637  025260  000400                  256.                    ;256. WORDS
7638  025262  070707                  070707                  ;FILL WITH 070707
7639
7640
7641                                  ;*FILL WRITE FROM BUFFER WITH NEXT SECTOR HEADER
7642
7643  025264  004037  033140          JSR     R0,@#FLHEAD     ;SAVE HEADER DATA IN WRFROM+<260.*2>
7644  025270  003400                  WRFROM+<260.*2>                 ;LOCATION WHERE SAVED
7645  025272  000004                  4                       ;NUMBER OF WORDS SAVED
7646  025274  010001                  10001                   ;FIRST DATA WORD
7647  025276  000001                  1                       ;SECOND DATA WORD
7648  025300  000000                  0                       ;THIRD DATA WORD
7649  025302  000000                  0                       ;FOURTH DATA WORD
7650
7651                                  ;*FILL WRITE FROM BUFFER WITH WITH NEXT SECTOR DATA
7652
7653  025304  004037  033164          JSR     R0,@#CLAREA     ;CLEAR 4 WORDS, FROM WRFROM+<268.*2>
7654  025310  003420                  WRFROM+<268.*2>                 ;STARTING FROM WRFROM+<268.*2>
7655  025312  000004                  4                       ;4 WORDS
7656  025314  070707                  70707                   ;FILL WITH 70707
7657
7658
7659                                  ;*NOW THE WRITE HEADER AND DATA COMMAND WILL BE FILLED
7660
7661  025316  004037  035276          JSR     R0,@#RUN        ;SETUP TO RUN FOR DATA COMMAND
7662  025322  000001                  1                       ;CYLINDER 1
7663  025324     000                  .BYTE   0               ;SECTOR 0
7664  025325     000                  .BYTE   0               ;TRACK 0
7665  025326  177364                  -264.-4                 ;WORD COUNT (DATA) = 264. +
7666                                                          ;4 HEADER WORDS
7667  025330  002370                  WRFROM                  ;BUS ADDRESS
7668                                                          ;STARTING ADDRESS OF DATA
7669                                                          ;BUFFER = WRFROM
7670  025332  000000                  0                       ;DO NOT INHIBIT BUS ADDRESS INCREMENT
7671  025334  010000                  FMT22                   ;16 BITS PER WORD FORMAT
7672                                                          ;DO NOT INHIBIT ECC CORRECTION
7673                                                          ;DO NOT INHIBIT HEADER COMPARE
7674  025336  002344                  WRIFOR                  ;GET READY TO DO A WR!FOR
7675                                                  ;WRITE HEADER AND DATA WITH 62 IN RHCS1
7676
7677
7678  025340  004737  033374          JSR     PC,@#CHECKT     ;CHECKS DVA,RDY,MOL,DPR,DRY AND VV = 1
7679                                                          ;AND THAT ALL STATUS BITS ARE = 0
7680  025344  104401  057465          TYPE    ,CPHALT         ;CANNOT CONTINUE TESTING IF NOT
7681  025350  000000                  HALT                    ;STOP TEST
7682
7683  025352  013777  004506  154606  MOV     @#RP4VEC,@RPVEC ;SET RP04 VECTOR ADDRESS
7684                                                          ;TO 'TIME1' IF P-CLOCK IS PRESENT
7685                                                          ;OR TO 'TIME2' IF P-CLOCK IS NOT THERE
7686                                                          ;'TIME' WILL ONLY SAVE
7687                                                          ;CURRENT CYLINDER ADDRESS
7688                                                          ;AND LOOK AHEAD REGISTERS
7689
```

```
7690
7691  025360  013746  002344                MOV     @#WRIFOR,-(SP)  ;GET READY TO MOVE COMMAND
7692  025364  052716  000101                BIS     #GO!IE,(SP)     ;GET READY TO SET GO AND
7693                                                                  ;ENABLE INTERRUPT
7694  025370  012677  154604                MOV     (SP)+,@RHCS1    ;GO WITH
7695                                                                   ;62 IN RHCS1 FOR WRITE HEADER AND DATA
7696                                                                  ;WITH INTERRUPT ENABLED
7697
7698                                        ;*ONE REVOLUTION=16670 MICRO SEC, ONE SECTOR=760 MICRO SEC
7699
7700  025374  104413                        WAT                     ;WAIT FOR RDY BIT TO SET
7701  025376  002200                        RHCS1                   ;WAIT FOR RHCS1 REGISTER
7702  025400  000200                        RDY                     ;WAIT FOR RDY BIT IN RHCS1 REGISTER
7703  025402  001725                        981.                    ;ALLOW 9810 MICRO SECONDS
7704  025404  001502                        834.                    ;RDY MUST SET BETWEEN
7705                                                                  ;1470 AND 18150 MICRO SECONDS
7706
7707                                        ;*NOW PREPARE FOR THE WRITE DATA COMMAND
7708
7709                                        ;*FILL WRITE FROM BUFFER WITH 256 OF 2000 AND 4 OF 2001
7710
7711  025406  004037  033164                JSR     R0,@#CLAREA     ;CLEAR 256. WORDS, FROM WRFROM
7712  025412  002370                        WRFROM                  ;STARTING FROM WRFROM
7713  025414  000400                        256.                    ;256. WORDS
7714  025416  002000                        2000                    ;FILL WITH 2000
7715
7716
7717  025420  004037  033164                JSR     R0,@#CLAREA     ;CLEAR 4 WORDS, FROM WRFROM+<256.*2>
7718  025424  003370                        WRFROM+<256.*2>                 ;STARTING FROM WRFROM+<256.*2>
7719  025426  000004                        4                       ;4 WORDS
7720  025430  002001                        2001                    ;FILL WITH 2001
7721
7722
7723                                        ;*FILL WRITE DATA COMMAND
7724
7725  025432  004037  035276                JSR     R0,@#RUN        ;SETUP TO RUN FOR DATA COMMAND
7726  025436  000001                        1                       ;CYLINDER 1
7727  025440     000                        .BYTE   0               ;SECTOR 0
7728  025441     000                        .BYTE   0               ;TRACK 0
7729  025442  177400                        -256.                   ;WORD COUNT = 256.
7730  025444  002370                        WRFROM                  ;BUS ADDRESS
7731                                                                  ;STARTING ADDRESS OF DATA
7732                                                                  ;BUFFER = WRFROM
7733  025446  000000                        0                       ;DO NOT INHIBIT BUS ADDRESS INCREMENT
7734  025450  010000                        FMT22                   ;16 BITS PER WORD FORMAT
7735                                                                  ;DO NOT INHIBIT ECC CORRECTION
7736                                                                  ;DO NOT INHIBIT HEADER COMPARE
7737  025452  002342                        WRIDAT                  ;GET READY TO DO A WRIDAT
7738                                                      ;WRITE DATA WITH 60 IN RHCS1
7739
7740
7741                                        ;*SAVE REGISTERS FOR COMPARISON AFTER ATTEMPTED
7742                                        ;*REGISTER MODIFICATION DURING A WRITE DATA
7743  025454  004037  033462                JSR     R0,@#SAVER      ;SAVE REGISTERS
7744  025460  002172                        RHWC                    ;RHWC IS THE FIRST REGISTER SAVED
7745  025462  004512                        SAVERE                  ;STARTING ADDRES OF WHERE
```

```
                                                        ;THE REGISTERS ARE SAVED
025464  000022                          18.            ;NUMBER OF REGISTERS
                                                        ;SAVED = 18.

025466  004737  033374                  JSR     PC,@#CHECKT     ;CHECKS DVA,RDY,MOL,DPR,DRY AND VV = 1
                                                        ;AND THAT ALL STATUS BITS ARE = 0
025472  104401  057465                  TYPE    ,CPHALT         ;CANNOT CONTINUE TESTING IF NOT
025476  000000                          HALT                    ;STOP TEST

025500  013777  004506  154460          MOV     @#RP4VEC,@RPVEC ;SET RP04 VECTOR ADDRESS
                                                        ;TO 'TIME1' IF P-CLOCK IS PRESENT
                                                        ;OR TO 'TIME2' IF P-CLOCK IS NOT THERE
                                                        ;'TIME' WILL ONLY SAVE
                                                        ;CURRENT CYLINDER ADDRESS
                                                        ;AND LOOK AHEAD REGISTERS

025506  013746  002342                  MOV     @#WRIDAT,-(SP)  ;GET READY TO MOVE COMMAND
025512  052716  000101                  BIS     #GO!IE,(SP)     ;GET READY TO SET GO AND
                                                        ;ENABLE INTERRUPT
025516  012677  154456                  MOV     (SP)+,@RHCS1    ;GO WITH
                                                         ;60 IN RHCS1 FOR WRITE DATA
                                                        ;WITH INTERRUPT ENABLED

025522  013700  004652                  MOV     @#TMP0,R0               ;SET R0 TO REG ADDRESS
025526  012730  002002                  MOV     #BIT1!BIT10,@(R0)+      ;ATTEMPT TO WRITE INTO A REGISTER
                                                                        ;DURING WRITE DATA
025532  010037  004652                  MOV     R0,@#TMP0               ;SAVE OFF NEW REG ADDRESS

                                        ;*NOW RMR MUST BE SET BUT THE COMPLETION OF
                                        ;*WRITE DATA MUST BE WAITED ON


025536  104413                          WAT                     ;WAIT FOR RDY BIT TO SET
025540  002200                          RHCS1                   ;WAIT FOR RHCS1 REGISTER
025542  000200                          RDY                     ;WAIT FOR RDY BIT IN RHCS1 REGISTER
025544  001725                          981.                    ;ALLOW 9810 MICRO SECONDS
025546  001502                          834.                    ;RDY MUST SET BETWEEN
                                                                ;1470 AND 18150 MICRO SECONDS

                                        ;*CHANGE SAVED REGISTERS TO EXPECTED VALUE

025550  004037  034204                  JSR     R0,@#CHREG      ;CHANGE BITS IN SAVED REGISTER
025554  002234                          RHCC                    ;CHANGE RHCC REGISTER

025556  000001                          1                       ;1 BIT/BITS TO BE CHANGED
025560  000001                          1                       ;NEW VALUE OF BIT0 IS 1
025562  000001                          BIT0                    ;CHANGE BIT0 BIT

025564  004037  034204                  JSR     R0,@#CHREG      ;CHANGE BITS IN SAVED REGISTER
025570  002200                          RHCS1                   ;CHANGE RHCS1 REGISTER

025572  000001                          1                       ;1 BIT/BITS TO BE CHANGED
025574  000001                          1                       ;NEW VALUE OF SC IS 1
025576  100000                          SC                      ;CHANGE SC BIT
```

```
7802  025600  004037  034204          JSR     R0,@#CHREG      ;CHANGE BITS IN SAVED REGISTER
7803  025604  002222                  RHDS1                   ;CHANGE RHDS1 REGISTER
7804
7805  025606  000002                  2                       ;2 BIT/BITS TO BE CHANGED
7806  025610  000001                  1                       ;NEW VALUE OF ATA IS 1
7807  025612  100000                  ATA                     ;CHANGE ATA BIT
7808  025614  000001                  1                       ;NEW VALUE OF ERR IS 1
7809  025616  040000                  ERR                     ;CHANGE ERR BIT
7810
7811  025620  053737  004644  004536  BIS     @#ATTENT,@#SAVERE+24 ;SET APPROPIATE 'ATA' BITS
7812                                                          ;FOR WORKING DRIVE IN
7813                                                          ;SAVED RHAS LOCATION
7814
7815  025626  004037  034204          JSR     R0,@#CHREG      ;CHANGE BITS IN SAVED REGISTER
7816  025632  002202                  RHER1                   ;CHANGE RHER1 REGISTER
7817
7818  025634  000001                  1                       ;1 BIT/BITS TO BE CHANGED
7819  025636  000001                  1                       ;NEW VALUE OF RMR IS 1
7820  025640  000004                  RMR                     ;CHANGE RMR BIT
7821
7822  025642  004037  033216          JSR     R0,@#FILLRE     ;MOV 1 INTO SAVED RHDST
7823  025646  002204                  RHDST                   ;SAVED REGISTER TO CHANGE
7824  025650  000001                  1                       ;DATA
7825
7826
7827  025652  004037  033216          JSR     R0,@#FILLRE     ;MOV WRFROM+<256.*2> INTO SAVED RHBA
7828  025656  002174                  RHBA                    ;SAVED REGISTER TO CHANGE
7829  025660  003370                  WRFROM+<256.*2>                 ;DATA
7830
7831
7832  025662  004037  033216          JSR     R0,@#FILLRE     ;MOV 0 INTO SAVED RHWC
7833  025666  002172                  RHWC                    ;SAVED REGISTER TO CHANGE
7834  025670  000000                  0                       ;DATA
7835
7836
7837                                  ;*COMPARE REGISTERS BEFORE WRITE DATA WITH REGISTERS
7838                                  ;*AFTER WRITE AND ATTEMPTED MODIFICATION OF REGISTER
7839
7840  025672  004037  034312          JSR     R0,@#COMREG     ;COMPARE SAVED REGISTERS WITH
7841                                                          ;PRESENT VALUE
7842  025676  004512                  SAVERE                  ;GOOD DATA SAVED IN 'SAVERE'
7843  025700  002254                  WC                      ;TEST DATA STARTING FROM 'RHWC'
7844  025702  000022                  18.                     ;18. REGISTERS TO BE COMPARED
7845  025704  025710                  2$                      ;RETURN TO 2$ ON ERROR
7846  025706  025730                  3$                      ;RETURN TO 3$ ON NO ERROR
7847
7848
7849  025710                  2$:
7850  025710  010046                  MOV     R0,-(SP)        ;;PUSH R0 ON STACK
7851  025712  013700  004652          MOV     @#TMP0,R0       ;GET REGISTER BEEING MODIFYED + 2 POINTER
7852  025716  014037  001122          MOV     -(R0),@#$BDADR  ;GET ADDRESS OF REGISTER BEING MODIFYED
7853  025722  104060                  ERROR   60              ;ATTEMPTING TO MODIFY REGISTER
7854  025724  012600                  MOV     (SP)+,R0        ;;POP STACK INTO R0
7855  025726  000207                  RTS     PC              ;DURING A WRITE COMMAND CAUSED
7856                                                          ;IMPROPER REGISTER GIVES WHAT SHOULD
7857                                                          ;GOOD DATA GIVES WHAT SHOULD
```

```
                                                        ;BE THERE
                                                        ;RECEIVED DATA GIVES WHAT WAS
                                                        ;THERE AFTER READ
                                    ;*CLEAR ALL ERROR FLAGS
025730                          3$:
                                    JSR     PC,@#CLDISK     ;SET R1-RHCS1, R2-RHCS2
                                                            ;R3-RHDS1, R4-RHER1
                                                            ;GIVE RH-11 INITIALIZE
                                                            ;SETUP UNIT NUMBER

                                    ;*FILL WRITE FROM BUFFER WITH EXPECTED DATA

025734  004037  033164              JSR     R0,@#CLAREA     ;CLEAR 256. WORDS, FROM WRFROM
025740  002370                      WRFROM                  ;STARTING FROM WRFROM
025742  000400                      256.                    ;256. WORDS
025744  002000                      2000                    ;FILL WITH 2000

025746  004037  033164              JSR     R0,@#CLAREA     ;CLEAR 4 WORDS, FROM WRFROM+<256.*2>
025752  003370                      WRFROM+<256.*2>                 ;STARTING FROM WRFROM+<256.*2>
025754  000004                      4                       ;4 WORDS
025756  002001                      2001                    ;FILL WITH 2001


                                    ;*NOW THE READ DATA COMMAND WILL BE FILLED

025760  004037  035276              JSR     R0,@#RUN        ;SETUP TO RUN FOR DATA COMMAND
025764  000001                      1                       ;CYLINDER 1
025766     000                      .BYTE   0               ;SECTOR 0
025767     000                      .BYTE   0               ;TRACK 0
025770  177374                      -260.                   ;WORD COUNT = 260.
025772  003434                      REINTO                  ;BUS ADDRESS
                                                            ;STARTING ADDRESS OF DATA
                                                            ;BUFFER = REINTO
025774  000000                      0                       ;DO NOT INHIBIT BUS ADDRESS INCREMENT
025776  014000                      ECI!FMT22                       ;16 BITS PER WORD FORMAT
                                                            ;INHIBIT ECC CORRECTION
                                                            ;DO NOT INHIBIT HEADER COMPARE
026000  002346                      READAT                  ;GET READY TO DO A READAT
                                                ;READ DATA WITH 70 IN RHCS1


                                    ;*NOW SAVE REGISTERS FOR COMPARISON AFTER READ DATA COMMAND
026002  004037  033462              JSR     R0,@#SAVER      ;SAVE REGISTERS
026006  002172                      RHWC                    ;RHWC IS THE FIRST REGISTER SAVED
026010  004512                      SAVERE                  ;STARTING ADDRES OF WHERE
                                                            ;THE REGISTERS ARE SAVED
026012  000022                      18.                     ;NUMBER OF REGISTERS
                                                            ;SAVED = 18.

026014  004737  033374              JSR     PC,@#CHECKT     ;CHECKS DVA,RDY,MOL,DPR,DRY AND VV = 1
                                                            ;AND THAT ALL STATUS BITS ARE = 0
026020  104401  057465              TYPE    ,CPHALT         ;CANNOT CONTINUE TESTING IF NOT
026024  000000                      HALT                    ;STOP TEST
```

```
 7914  026026  013777  004506  154132          MOV     @#RP4VEC,@RPVEC ;SET RP04 VECTOR ADDRESS
 7915                                                                    ;TO 'TIME1' IF P-CLOCK IS PRESENT
 7916                                                                    ;OR TO 'TIME2' IF P-CLOCK IS NOT THERE
 7917                                                                    ;'TIME' WILL ONLY SAVE
 7918                                                                    ;CURRENT CYLINDER ADDRESS
 7919                                                                    ;AND LOOK AHEAD REGISTERS
 7920
 7921
 7922  026034  013746  002346                  MOV     @#READAT,-(SP)  ;GET READY TO MOVE COMMAND
 7923  026040  052716  000101                  BIS     #GO!IE,(SP)     ;GET READY TO SET GO AND
 7924                                                                    ;ENABLE INTERRUPT
 7925  026044  012677  154130                  MOV     (SP)+,@RHCS1    ;GO WITH
 7926                                                                     ;70 IN RHCS1 FOR READ DATA
 7927                                                                    ;WITH INTERRUPT ENABLED
 7928  026050  011100                          MOV     @R1,R0          ;SAVE RHCS1 DURING ABOVE OPERATION
 7929  026052  011305                          MOV     @R3,R5          ;SAVE RHDS1 DURING ABOVE OPERATION
 7930
 7931
 7932  026054  104413                          WAT                     ;WAIT FOR RDY BIT TO SET
 7933  026056  002200                          RHCS1                   ;WAIT FOR RHCS1 REGISTER
 7934  026060  000200                          RDY                     ;WAIT FOR RDY BIT IN RHCS1 REGISTER
 7935  026062  001725                          981.                    ;ALLOW 9810 MICRO SECONDS
 7936  026064  001502                          834.                    ;RDY MUST SET BETWEEN
 7937                                                                    ;1470 AND 18150 MICRO SECONDS
 7938
 7939                                          ;*COMPARE CONTENTS OF RHCS1 AND RHDS1 ALREADY SAVED IN
 7940                                          ;*R0 AND R5 IMMEDIATELY AFTER GO IS ISSUED
 7941
 7942  026066  013746  002346                  MOV     @#READAT,-(SP)  ;SAVE COMMAND
 7943  026072  052716  004101                  BIS     #IE!DVA!GO,(SP) ;INCLUDE IE!DVA!GO
 7944  026076  011637  001124                  MOV     (SP),@#$GDDAT   ;SAVE FOR PRINTOUT
 7945  026102  022600                          CMP     (SP)+,R0        ;DURING ABOVE OPERATION ONLY IE!DVA!GO
 7946                                                                    ;AND COMMAND SHOULD BE SET
 7947  026104  001405                          BEQ     70$             ;BRANCH IF GOOD
 7948  026106  010037  001126                  MOV     R0,@#$BDDAT     ;BAD DATA
 7949  026112  010137  004500                  MOV     R1,@#REGADR     ;FAILING REGISTER RHCS1
 7950  026116  104021                          ERROR   21              ;DURING ABOVE OPERATION ONLY
 7951                                                                    ;COMMAND AND IE!DVA!GO SHOULD BE SET
 7952  026120  012746  010500          70$:    MOV     #MOL!DPR!VV,-(SP)       ;SAVE BITS SET DURING OPERATION IN RHDS1
 7953  026124  011637  001124                  MOV     (SP),@#$GDDAT   ;SAVE FOR PRINTOUT
 7954  026130  022605                          CMP     (SP)+,R5        ;DURING ABOVE OPERATION ONLY MOL!DPR!VV
 7955                                                                    ;SHOULD BE SET
 7956  026132  001405                          BEQ     72$             ;BRANCH IF GOOD
 7957  026134  010537  001126                  MOV     R5,@#$BDDAT     ;BAD DATA
 7958  026140  010337  004500                  MOV     R3,@#REGADR     ;FAILING REGISTER RHDS1
 7959  026144  104063                          ERROR   63              ;DURING ABOVE OPERATION ONLY
 7960                                                                    ;MOL!DPR!VV SHOULD BE SET
 7961  026146                          72$:
 7962
 7963                                          ;*NOW CHANGE SAVED REGISTERS TO EXPECTED VALUES
 7964
 7965  026146  004037  033216                  JSR     R0,@#FILLRE     ;MOV 0 INTO SAVED RHWC
 7966  026152  002172                          RHWC                    ;SAVED REGISTER TO CHANGE
 7967  026154  000000                          0                       ;DATA
 7968
 7969
```

```
7970
7971  026156  004037  033216                  JSR     R0,@#FILLRE     ;MOV REINTO+<260.*2> INTO SAVED RHBA
7972  026162  002174                          RHBA                    ;SAVED REGISTER TO CHANGE
7973  026164  004444                          REINTO+<260.*2>                 ;DATA
7974
7975
7976  026166  004037  033216                  JSR     R0,@#FILLRE     ;MOV 2 INTO SAVED RHDST
7977  026172  002204                          RHDST                   ;SAVED REGISTER TO CHANGE
7978  026174  000002                          2                       ;DATA
7979
7980
7981                                          ;*NOW COMPARE REGISTERS BEFORE READ DATA WITH
7982                                          ;*AFTER COMMAND
7983
7984  026176  004037  034312                  JSR     R0,@#COMREG     ;COMPARE SAVED REGISTERS WITH
7985                                                                  ;PRESENT VALUE
7986  026202  004512                          SAVERE                  ;GOOD DATA SAVED IN 'SAVERE'
7987  026204  002254                          WC                      ;TEST DATA STARTING FROM 'RHWC'
7988  026206  000022                          18.                     ;18. REGISTERS TO BE COMPARED
7989  026210  026214                          4$                      ;RETURN TO 4$ ON ERROR
7990  026212  026220                          5$                      ;RETURN TO 5$ ON NO ERROR
7991
7992
7993  026214  104033              4$:         ERROR   33              ;READ DATA CAUSED IMPROPER REGISTER
7994  026216  000207                          RTS     PC              ;CHANGE
7995                                                                  ;GOOD DATA GIVES WHAT SHOULD BE THERE
7996                                                                  ;RECEIVED DATA GIVES WHAT WAS THERE AFTER COMMAND
7997                                          ;*NOW READ INTO BUFFER WILL BE CHECKED TO SEE THAT READ
7998                                          ;*WAS GOOD
7999  026220                      5$:
8000
8001  026220  004037  035342                  JSR     R0,@#COMPAR     ;COMPARE TWO BLOCKS OF MEMORY
8002  026224  002370                          WRFROM                  ;GOOD DATA STARTS FROM WRFROM
8003  026226  003434                          REINTO                  ;TEST DATA STARTS FROM REINTO
8004  026230  000400                          256.                    ;256., WORDS TO BE COMPARED
8005  026232  026236                          6$                      ;RETURN TO 6$ ON ERROR
8006  026234  026242                          ST28                    ;RETURN TO ST28 ON NO ERROR
8007
8008
8009  026236  104034              6$:         ERROR   34              ;READ DATA ERROR AFTER A WRITE DATA
8010  026240  000207                          RTS     PC              ;WITH REGISTER MODIFICATION
8011                                                                  ;WITHIN THE WRITE DATA
8012                                          ;*IF ALL 7 REGISTERS NOT COMPLETE THEN REPEAT
8013  026242  005337  004660      ST28:       DEC     @#TMP5          ;COUNT DOWN
8014  026246  001002                          BNE     1$              ;BRANCH IF 7 NOT DONE
8015  026250  000137  026260                  JMP     TST31           ;JUMP TO NEXT TEST
8016  026254  000137  025226      1$:         JMP     ST24            ;JUMP TO BEGINING OF TEST
8017
```

```
8018
8019                                    ;;*****************************************************************
8020                                    ;*TEST 31        REGISTER MODIFICATION REFUSED - RHER1  (BIT #2)RMR
8021
8022                                    ;*      A READ DATA COMMAND IS GIVEN TO CYLINDER 0, SECTOR 0
8023                                    ;*      TRACK 0. IMMEDIATELY AFTER GO RHAS IS WRITTEN INTO
8024                                    ;*      WITH ALL ONES RMR BIT #2 IN RHER SHOULS NOT SET
8025
8026                                    ;;*****************************************************************
8027  026260  000004                   TST31:  SCOPE
8028  026262  012706  001000                   MOV     #STACK,SP       ;RESET STACK
8029  026266  012737  000031  004504           MOV     #31,@#TSTNM     ;SAVE TEST NUMBER
8030
8031  026274  004737  033314                   JSR     PC,@#CLDISK     ;SET R1-RHCS1, R2-RHCS2
8032                                                                    ;R3-RHDS1, R4-RHER1
8033                                                                    ;GIVE RH-11 INITIALIZE
8034                                                                    ;SETUP UNIT NUMBER
8035
8036                                    ;*FILL WRITE FROM BUFFER WITH EXPECTED DATA
8037
8038  026300  004037  033164                   JSR     R0,@#CLAREA     ;CLEAR 256. WORDS, FROM WRFROM
8039  026304  002370                           WRFROM                  ;STARTING FROM WRFROM
8040  026306  000400                           256.                    ;256. WORDS
8041  026310  000000                           0                       ;FILL WITH 0
8042
8043                                    ;*FILL READ INTM BUFFER WITH ALL ONES
8044
8045  026312  004037  033164                   JSR     R0,@#CLAREA     ;CLEAR 256. WORDS, FROM REINTO
8046  026316  003434                           REINTO                  ;STARTING FROM REINTO
8047  026320  000400                           256.                    ;256. WORDS
8048  026322  177777                           -1                      ;FILL WITH -1
8049
8050                                    ;*NOW THE READ DATA COMMAND WILL BE FILLED
8051
8052  026324  004037  035276                   JSR     R0,@#RUN        ;SETUP TO RUN FOR DATA COMMAND
8053  026330  000000                           0                       ;CYLINDER 0
8054  026332     000                    .BYTE   0                       ;SECTOR 0
8055  026333     000                    .BYTE   0                       ;TRACK 0
8056  026334  177400                           -256.                   ;WORD COUNT = 256.
8057  026336  003434                           REINTO                  ;BUS ADDRESS
8058                                                                    ;STARTING ADDRESS OF DATA
8059                                                                    ;BUFFER = REINTO
8060  026340  000000                           0                       ;DO NOT INHIBIT BUS ADDRESS INCREMENT
8061  026342  014000                           ECI!FMT22                       ;16 BITS PER WORD FORMAT
8062                                                                    ;INHIBIT ECC CORRECTION
8063                                                                    ;DO NOT INHIBIT HEADER COMPARE
8064  026344  002346                           READAT                  ;GET READY TO DO A READAT
8065                                                        ;READ DATA WITH 70 IN RHCS1
8066
8067                                    ;*NOW SAVE REGISTERS FOR COMPARISON AFTER READ DATA COMMAND
8068  026346  004037  033462                   JSR     R0,@#SAVER      ;SAVE REGISTERS
8069  026352  002172                           RHWC                    ;RHWC IS THE FIRST REGISTER SAVED
8070  026354  004512                           SAVERE                  ;STARTING ADDRES OF WHERE
8071                                                                    ;THE REGISTERS ARE SAVED
8072  026356  000022                           18.                     ;NUMBER OF REGISTERS
8073                                                                    ;SAVED = 18.
```

```
8074
8075  026360  004737  033374                  JSR     PC,@#CHECKT     ;CHECKS DVA,RDY,MOL,DPR,DRY AND VV = 1
8076                                                                    ;AND THAT ALL STATUS BITS ARE = 0
8077  026364  104401  057465                  TYPE    ,CPHALT         ;CANNOT CONTINUE TESTING IF NOT
8078  026370  000000                          HALT                    ;STOP TEST
8079
8080  026372  013777  004506  153566          MOV     @#RP4VEC,@RPVEC ;SET RP04 VECTOR ADDRESS
8081                                                                    ;TO 'TIME1' IF P-CLOCK IS PRESENT
8082                                                                    ;OR TO 'TIME2' IF P-CLOCK IS NOT THERE
8083                                                                    ;'TIME' WILL ONLY SAVE
8084                                                                    ;CURRENT CYLINDER ADDRESS
8085                                                                    ;AND LOOK AHEAD REGISTERS
8086
8087
8088
8089  026400  013746  002346                  MOV     @#READAT,-(SP)  ;GET READY TO MOVE COMMAND
8090  026404  052716  000101                  BIS     #GO!IE,(SP)     ;GET READY TO SET GO AND
8091                                                                    ;ENABLE INTERRUPT
8092  026410  012677  153564                  MOV     (SP)+,@RHCS1    ;GO WITH
8093                                                                    ;70 IN RHCS1 FOR READ DATA
8094                                                                    ;WITH INTERRUPT ENABLED
8095  026414  011100                          MOV     @R1,R0          ;SAVE RHCS1 DURING ABOVE OPERATION
8096  026416  011305                          MOV     @R3,R5          ;SAVE RHDS1 DURING ABOVE OPERATION
8097
8098  026420  012777  177777  153570          MOV     #-1,@RHAS       ;WRITE INTO RHAS THIS SHOULD
8099                                                                    ;NOT SET RMR
8100
8101                                          ;*TIME IS NOT IMPORTANT
8102
8103  026426  104413                          WAT                     ;WAIT FOR RDY BIT TO SET
8104  026430  002200                          RHCS1                   ;WAIT FOR RHCS1 REGISTER
8105  026432  000200                          RDY                     ;WAIT FOR RDY BIT IN RHCS1 REGISTER
8106  026434  003326                          1750.                   ;ALLOW 17500 MICRO SECONDS
8107  026436  000175                          125.                    ;RDY MUST SET BETWEEN
8108                                                                    ;16250 AND 18750 MICRO SECONDS
8109
8110                                          ;*COMPARE CONTENTS OF RHCS1 AND RHDS1 ALREADY SAVED IN
8111                                          ;*R0 AND R5 IMMEDIATELY AFTER GO IS ISSUED
8112
8113  026440  013746  002346                  MOV     @#READAT,-(SP)  ;SAVE COMMAND
8114  026444  052716  004101                  BIS     #IE!DVA!GO,(SP) ;INCLUDE IE!DVA!GO
8115  026450  011637  001124                  MOV     (SP),@#$GDDAT   ;SAVE FOR PRINTOUT
8116  026454  022600                          CMP     (SP)+,R0        ;DURING ABOVE OPERATION ONLY IE!DVA!GO
8117                                                                    ;AND COMMAND SHOULD BE SET
8118  026456  001405                          BEQ     64$             ;BRANCH IF GOOD
8119  026460  010037  001126                  MOV     R0,@#$BDDAT     ;BAD DATA
8120  026464  010137  004500                  MOV     R1,@#REGADR     ;FAILING REGISTER RHCS1
8121  026470  104021                          ERROR   21              ;DURING ABOVE OPERATION ONLY
8122                                                                    ;COMMAND AND IE!DVA!GO SHOULD BE SET
8123  026472  012746  010500          64$:    MOV     #MOL!DPR!VV,-(SP)       ;SAVE BITS SET DURING OPERATION IN RHDS1
8124  026476  011637  001124                  MOV     (SP),@#$GDDAT   ;SAVE FOR PRINTOUT
8125  026502  022605                          CMP     (SP)+,R5        ;DURING ABOVE OPERATION ONLY MOL!DPR!VV
8126                                                                    ;SHOULD BE SET
8127  026504  001405                          BEQ     66$             ;BRANCH IF GOOD
8128  026506  010537  001126                  MOV     R5,@#$BDDAT     ;BAD DATA
8129  026512  010337  004500                  MOV     R3,@#REGADR     ;FAILING REGISTER RHDS1
```

```
8130  026516  104063                    ERROR   63              ;DURING ABOVE OPERATION ONLY
8131                                                            ;MOL!DPR!VV SHOULD BE SET
8132  026520                        66$:
8133
8134                                    ;*NOW CHANGE SAVED REGISTERS TO EXPECTED VALUES
8135
8136  026520  004037  033216            JSR     R0,@#FILLRE     ;MOV 0 INTO SAVED RHWC
8137  026524  002172                    RHWC                    ;SAVED REGISTER TO CHANGE
8138  026526  000000                    0                       ;DATA
8139
8140
8141  026530  004037  033216            JSR     R0,@#FILLRE     ;MOV REINTO+<256.*2> INTO SAVED RHBA
8142  026534  002174                    RHBA                    ;SAVED REGISTER TO CHANGE
8143  026536  004434                    REINTO+<256.*2>                 ;DATA
8144
8145
8146  026540  004037  033216            JSR     R0,@#FILLRE     ;MOV 1 INTO SAVED RHDST
8147  026544  002204                    RHDST                   ;SAVED REGISTER TO CHANGE
8148  026546  000001                    1                       ;DATA
8149
8150
8151  026550  004037  033216            JSR     R0,@#FILLRE     ;MOV 0 INTO SAVED RHCC
8152  026554  002234                    RHCC                    ;SAVED REGISTER TO CHANGE
8153  026556  000000                    0                       ;DATA
8154
8155
8156                                    ;*NOW COMPARE REGISTERS BEFORE READ DATA WITH
8157                                    ;*AFTER COMMAND
8158
8159  026560  004037  034312            JSR     R0,@#COMREG     ;COMPARE SAVED REGISTERS WITH
8160                                                            ;PRESENT VALUE
8161  026564  004512                    SAVERE                  ;GOOD DATA SAVED IN 'SAVERE'
8162  026566  002254                    WC                      ;TEST DATA STARTING FROM 'RHWC'
8163  026570  000022                    18.                     ;18. REGISTERS TO BE COMPARED
8164  026572  026576                    1$                      ;RETURN TO 1$ ON ERROR
8165  026574  026602                    2$                      ;RETURN TO 2$ ON NO ERROR
8166
8167
8168  026576  104033            1$:     ERROR   33              ;READ DATA CAUSED IMPROPER REGISTER
8169  026600  000207                    RTS     PC              ;CHANGE
8170                                                            ;GOOD DATA GIVES WHAT SHOULD BE THERE
8171                                                            ;RECEIVED DATA GIVES WHAT WAS THERE AFTER COMMAND
8172                                    ;*NOW READ INTO BUFFER WILL BE CHECKED TO SEE THAT READ
8173                                    ;*WAS GOOD
8174  026602                        2$:
8175
8176  026602  004037  035342            JSR     R0,@#COMPAR     ;COMPARE TWO BLOCKS OF MEMORY
8177  026606  002370                    WRFROM                  ;GOOD DATA STARTS FROM WRFROM
8178  026610  003434                    REINTO                  ;TEST DATA STARTS FROM REINTO
8179  026612  000400                    256.                    ;256., WORDS TO BE COMPARED
8180  026614  026620                    3$                      ;RETURN TO 3$ ON ERROR
8181  026616  026624                    4$                      ;RETURN TO 4$ ON NO ERROR
8182
8183
8184  026620  104034            3$:     ERROR   34              ;READ DATA ERROR AFTER WRITING INTO
8185  026622  000207                    RTS     PC              ;RHAS DURING READ
```

```
 8186
 8187  026624                          4$:
```

```
                                        ;;*****************************************************************
                                        ;*TEST 32        ILLEGAL FUNCTION RHER1 - (BIT #0,ILF)

                                        ;*      THIS WILL CALCULATE EVERY ILLEGAL FUNCTION
                                        ;*      BETWEEN 0 AND 77. EACH TIME AN ILLEGAL FUNCTION
                                        ;*      IS FORMED IT WILL BE STORED IN ILLEGAL THEN
                                        ;*      EXECUTION OF ILLEGAL
                                        ;*      WILL BE ATTEMPTED AND RESULTS CHECKED

                                        ;;*****************************************************************
026624  000004                  TST32:  SCOPE
026626  012706  001000                  MOV     #STACK,SP       ;RESET STACK
026632  012737  000032  004504          MOV     #32,@#TSTNM     ;SAVE TEST NUMBER

026640  004737  033314                  JSR     PC,@#CLDISK     ;SET R1-RHCS1, R2-RHCS2
                                                                ;R3-RHDS1, R4-RHER1
                                                                ;GIVE RH-11 INITIALIZE
                                                                ;SETUP UNIT NUMBER

026644  005737  004640                  TST     @#RH70          ;RH70 CONTROLLER ?
026650  001402                          BEQ     30$             ;SKIP NEXT IF NOT = 1
026652  000137  027452                  JMP     TST33           ;IF SO SKIP THIS TEST ----------------)
026656                          30$:

                                        ;*GENERATE ILLEGAL FUNCTION

026656  005037  001200                  CLR     @#$TMP1         ;GET READY TO MAKE ILLEGAL FUNCTION
026662  012700  002322          1$:     MOV     #FUTABL,R0      ;GET POINTER TO BEGINNING OF COMMANDS
026666  012705  000021                  MOV     #17.,R5         ;COUNTER (17 GOOD FUNCTIONS)
026672  023720  001200          2$:     CMP     @#$TMP1,(R0)+   ;IS THIS A LEGAL FUNCTION
026676  001004                          BNE     3$              ;BRANCH IF NOT LEGAL
026700  062737  000002  001200          ADD     #2,@#$TMP1      ;MAKE ANOTHER FUNCTION
026706  000765                          BR      1$              ;GET READY TO TEST NEW FUNCTION
026710  005305                  3$:     DEC     R5              ;NOT LEGAL SO DECREMENT COUNTER
026712  001367                          BNE     2$              ;BRANCH IF 17 NOT DONE
026714  032737  000100  001200          BIT     #100,@#$TMP1    ;ALL BITS UP TO BIT #5 COMPARED?
026722  001001                          BNE     20$             ;BRANCH OUT IF DONE
026724  000402                          BR      19$             ;BRANCH TO CONTINUE
026726  000137  027452          20$:    JMP     @#7$            ; DONE
026732  013737  001200  002364  19$:    MOV     @#$TMP1,@#ILLEGL ;AN ILLEGAL FUNCTION IS FOUND
026740  062737  000002  001200          ADD     #2,@#$TMP1      ;GET READY FOR NEW FUNCTION NEXT TIME

                                        ;*ILLEGAL FUNCTION HAS BEEN FOUND
                                        ;*IT IS IN 'ILLEGL'
026746  012737  026754  001110          MOV     #4$,@#$LPERR    ;ERROR RETURN POINT

                                        ;*SAVE REGISTERS FOR COMPARISON AFTER GO
026754                          4$:

026754  004737  033314                  JSR     PC,@#CLDISK     ;SET R1-RHCS1, R2-RHCS2
                                                                ;R3-RHDS1, R4-RHER1
                                                                ;GIVE RH-11 INITIALIZE
                                                                ;SETUP UNIT NUMBER
026760  005077  153206                  CLR     @RHWC           ;CLEAR WORD COUNT
026764  005077  153204                  CLR     @RHBA           ;CLEAR BUS ADDRESS
```

```
8244  026770  023727  002364  000050          CMP     @#ILLEGL,#50    ;50 AND HIGHER FUNCTIONS ARE DATA
8245                                                                  ;FUNCTIONS WHICH WILL SET MXF AND TRE
8246  026776  103014                          BHIS    13$             ;BRANCH IF ILLEGL IS HIGHER THAN 50
8247  027000  012737  100000  027354          MOV     #SC,@#11$+12    ;EXPECTED VALUE OF RHCS1 SHOULD HAVE
8248                                                                  ;ONLY SC ADDED
8249  027006  005037  027376                  CLR @#12$+12            ;EXPECTED VALUE OF RHCS2 SHOULD HAVE
8250                                                                  ;NOTHING ADDED
8251  027012  005037  027402                  CLR     @#12$+16        ;NO BITS TO BE CLEARED IN RHCS2
8252  027016  005037  027412                  CLR     @#15$+6         ;RHBA SHOULD BE 0
8253  027022  005037  027422                  CLR     @#16$+6         ;CLEAR SAVED RHWC
8254  027026  000500                          BR      14$             ;BRANCH
8255  027030  022737  000064  002364  13$:    CMP     #64,@#ILLEGL    ;IS FUNCTION 64
8256  027036  001020                          BNE     17$             ;BRANCH IF NOT
8257  027040  012737  140000  027354          MOV     #SC!TRE,@#11$+12;SAVED RHCS1 SHOULD HAVE SC AND TRE
8258  027046  012737  000204  027412          MOV     #204,@#15$+6    ;RHBA SHOULD HAVE 204
8259  027054  012737  000102  027422          MOV     #102,@#16$+6    ;RHWC SHOULD HAVE 102
8260  027062  012737  001200  027376          MOV     #MXF!OR,@#12$+12;RHCS2 SHOULD HAVE MXF AND OR
8261  027070  012737  000100  027402          MOV     #IR,@#12$+16    ;RHCS2 SHOULD HAVE IR CLEARED
8262  027076  000454                          BR      14$             ;BRANCH
8263  027100  022737  000066  002364  17$:    CMP     #66,@#ILLEGL    ;IS FUNCTION 66
8264  027106  001030                          BNE     18$             ;BRANCH IF NOT
8265  027110  012777  177672  153054          MOV     #-70.,@RHWC     ; MOVE 70 INTO RHWC
8266  027116  012777  002370  153050          MOV     #WRFROM,@RHBA   ;FILL RHBA WITH WRFROM
8267  027124  012737  140000  027354          MOV     #SC!TRE,@#11$+12;SAVED RHCS1
8268  027132  012737  002164  027412          MOV     #WRFROM-<66.*2>,15$+6;RHBA
8269  027140  012737  177774  027422          MOV     #-4.,16$+6      ;SAVED RHWC
8270  027146  012737  001200  027376          MOV     #MXF!OR,@#12$+12 ;SAVED RHCS2
8271  027154  005037  027402                  CLR     @#12$+16        ;RHCS2
8272  027160  012737  000100  027402          MOV     #IR,@#12$+16    ;RHCS2 SHOULD HAVE IR CLEARED
8273  027166  000420                          BR      14$             ;BRANCH
8274  027170  005077  152776          18$:    CLR     @RHWC           ;CLEAR RHWC
8275  027174  005077  152774                  CLR     @RHBA           ;CLEAR RHBA
8276  027200  012737  140000  027354          MOV     #SC!TRE,@#11$+12;RHCS1 SHOULD HAVE SC AND TRE
8277  027206  005037  027412                  CLR     @#15$+6         ;RHBA
8278  027212  005037  027422                  CLR     @#16$+6         ;RHWC
8279  027216  012737  001000  027376          MOV     #MXF,@#12$+12   ;RHCS2
8280  027224  005037  027402                  CLR     @#12$+16        ;RHCS2
8281  027230                          14$:
8282  027230  004037  033462                  JSR     R0,@#SAVER      ;SAVE REGISTERS
8283  027234  002172                          RHWC                    ;RHWC IS THE FIRST REGISTER SAVED
8284  027236  004512                          SAVERE                  ;STARTING ADDRES OF WHERE
8285                                                                  ;THE REGISTERS ARE SAVED
8286  027240  000022                          18.                     ;NUMBER OF REGISTERS
8287                                                                  ;SAVED = 18.
8288
8289  027242  004737  033374                  JSR     PC,@#CHECKT     ;CHECKS DVA,RDY,MOL,DPR,DRY AND VV = 1
8290                                                                  ;AND THAT ALL STATUS BITS ARE = 0
8291  027246  104401  057465                  TYPE    ,CPHALT         ;CANNOT CONTINUE TESTING IF NOT
8292  027252  000000                          HALT                    ;STOP TEST
8293  027254  013746  002364                  MOV     @#ILLEGL,-(SP)  ;GET ILLEGAL FUNCTION
8294  027260  052716  000101                  BIS     #GO!IE,(SP)     ;INCLUDE IE AND GO
8295  027264  012611                          MOV     (SP)+,@R1       ;GO TO RHCS1 WITH ILLEGAL FUNCTION
8296
8297  027266  104413                          WAT                     ;WAIT FOR RDY BIT TO SET
8298  027270  002200                          RHCS1                   ;WAIT FOR RHCS1 REGISTER
8299  027272  000200                          RDY                     ;WAIT FOR RDY BIT IN RHCS1 REGISTER
```

```
                                        908.                    ;ALLOW 9080 MICRO SECONDS
                                        907.                    ;RDY MUST SET BETWEEN
                                                                ;10 AND 18150 MICRO SECONDS

                                        ;*CHANGE SAVED REGISTERS TO EXPECTED VALUE

                                        JSR     R0,@#CHREG      ;CHANGE BITS IN SAVED REGISTER
                                        RHER1                   ;CHANGE RHER1 REGISTER

                                        1                       ;1 BIT/BITS TO BE CHANGED
                                        1                       ;NEW VALUE OF ILF IS 1
                                        ILF                     ;CHANGE ILF BIT

                                        JSR     R0,@#CHREG      ;CHANGE BITS IN SAVED REGISTER
                                        RHDS1                   ;CHANGE RHDS1 REGISTER

                                        2                       ;2 BIT/BITS TO BE CHANGED
                                        1                       ;NEW VALUE OF ATA IS 1
                                        ATA                     ;CHANGE ATA BIT
                                        1                       ;NEW VALUE OF ERR IS 1
                                        ERR                     ;CHANGE ERR BIT

                                        BIS     @#ATTENT,@#SAVERE+24 ;SET APPROPIATE 'ATA' BITS
                                                                ;FOR WORKING DRIVE IN
                                                                ;SAVED RHAS LOCATION

                                        ;*RHCS1 WILL HAVE SC AND TRE ADDED IF FUNCTION IS GREATER THAN 50
                                11$:

                                        JSR     R0,@#CHREG      ;CHANGE BITS IN SAVED REGISTER
                                        RHCS1                   ;CHANGE RHCS1 REGISTER

                                        1                       ;1 BIT/BITS TO BE CHANGED
                                        1                       ;NEW VALUE OF SC IS 1
                                        SC                      ;CHANGE SC BIT
                                        BIS     @#ILLEGL,@#SAVERE+6;INCLUDE ILLEGAL FUNCTION
                                                                ;IN RHCS1
                                        ;*RHCS2 WILL HAVE NOTHING ADDED IF FUNCTION IS LESS THAN 50
                                12$:

                                        JSR     R0,@#CHREG      ;CHANGE BITS IN SAVED REGISTER
                                        RHCS2                   ;CHANGE RHCS2 REGISTER

                                        2                       ;2 BIT/BITS TO BE CHANGED
                                        1                       ;NEW VALUE OF MXF IS 1
                                        MXF                     ;CHANGE MXF BIT
                                        0                       ;NEW VALUE OF IR IS 0
                                        IR                      ;CHANGE IR BIT
                                15$:

                                        JSR     R0,@#FILLRE     ;MOV 0 INTO SAVED RHBA
                                        RHBA                    ;SAVED REGISTER TO CHANGE
                                        0                       ;DATA

                                16$:
```

```
 8356  027414  004037  033216            JSR     R0,@#FILLRE     ;MOV 0 INTO SAVED RHWC
 8357  027420  002172                    RHWC                    ;SAVED REGISTER TO CHANGE
 8358  027422  000000                    0                       ;DATA
 8359
 8360
 8361                                    ;*NOW COMPARE REGISTERS AFTER GIVING AN ILLEGAL COMMAND
 8362
 8363  027424  004037  034312            JSR     R0,@#COMREG     ;COMPARE SAVED REGISTERS WITH
 8364                                                            ;PRESENT VALUE
 8365  027430  004512                    SAVERE                  ;GOOD DATA SAVED IN 'SAVERE'
 8366  027432  002254                    WC                      ;TEST DATA STARTING FROM 'RHWC'
 8367  027434  000022                    18.                     ;18. REGISTERS TO BE COMPARED
 8368  027436  027442                    5$                      ;RETURN TO 5$ ON ERROR
 8369  027440  027446                    6$                      ;RETURN TO 6$ ON NO ERROR
 8370
 8371
 8372  027442  104051          5$:       ERROR   51              ;GIVING ILLEGAL FUNCTION CAUSED
 8373  027444  000207                    RTS     PC              ;IMPROPER REGISTER CHANGE
 8374                                                            ;GOOD DATA GIVES WHAT
 8375                                                            ;SHOULD BE THERE
 8376                                                            ;RECEIVED DATA GIVES REGISTER
 8377                                                            ;CONTENTS AFTER ILLEGAL
 8378                                                            ;FUNCTION WA GIVEN
 8379  027446  000137  026662    6$:     JMP     @#1$            ;BRANCH FOR NEXT FUNCTION
 8380  027452                    7$:
 8381
 8382  027452                    10$:
 8383
 8384
 8385
 8386
```

```
8387
8388                                    ;;*****************************************************************
8389                                    ;*TEST 33       OPERATION INCOMPLETE - RHER1(BIT #13)OPI
8390                                    ;*      A WRITE HEADER AND DATA COMMAND IS GIVEN
8391                                    ;*      CYLINDER 0 SECTOR 1 TRACK 0 KEYS 0 DATA 177777
8392                                    ;*      WORDCOUNT 260
8393                                    ;*
8394                                    ;*      AFTER GO IS GIVEN THEN THREE INDEX PULSES ARE
8395                                    ;*      GIVEN. THIS SHOULD BRING OPI HIGH
8396                                    ;;*****************************************************************
8397  027452  000004                    TST33:  SCOPE
8398  027454  012706  001000                    MOV     #STACK,SP       ;RESET STACK
8399  027460  012737  000033  004504            MOV     #33,@#TSTNM     ;SAVE TEST NUMBER
8400
8401  027466  004737  033314                    JSR     PC,@#CLDISK     ;SET R1-RHCS1, R2-RHCS2
8402                                                                    ;R3-RHDS1, R4-RHER1
8403                                                                    ;GIVE RH-11 INITIALIZE
8404                                                                    ;SETUP UNIT NUMBER
8405
8406                                            ;*THESE ARE REGULAR SETUPS
8407  027472  012777  177374  152472            MOV     #-260.,@RHWC    ;256 DATA WORDS 4 HEADER WORDS
8408  027500  012700  002370                    MOV     #WRFROM,R0      ;THESE TWO INSTRUCTIONS GETS
8409  027504  010077  152464                    MOV     R0,@RHBA        ;ADDR. OF WRFROM BUFFER INTO R0 AND
8410                                                                    ;BUS ADDRESS REGISTER
8411  027510  012710  010000                    MOV     #FMT22,(R0);    ;FORMAT=16 BIT WORDS
8412                                                                    ;CYLINDER=0
8413  027514  012720  000001                    MOV     #1,(R0)+        ;TRACK=0, SECTOR=1, KEYS=0
8414  027520  005020                            CLR     (R0)+           ;KEY1=0
8415  027522  005020                            CLR     (R0)+           ;KEY2=0
8416  027524  012705  000400                    MOV     #256.,R5        ;COUNTER
8417
8418                                            ;*SETUP DATA, WRITE HEADER & DATA, AND FORMAT OF THE WRITE
8419  027530  012720  177777            1$:     MOV     #-1,(R0)+       ;MOVE ALL ONES FOR DATA
8420  027534  005305                            DEC     R5
8421  027536  001374                            BNE     1$              ;BRANCH IF DATA NOT COMPLETE
8422  027540  012777  000001  152436            MOV     #1,@RHDST       ;TRACK=0 SECTOR=1
8423  027546  004737  033374                    JSR     PC,@#CHECKT     ;CHECKS DVA,RDY,MOL,DPR,DRY AND VV = 1
8424                                                                    ;AND THAT ALL STATUS BITS ARE = 0
8425  027552  104401  057465                    TYPE    ,CPHALT         ;CANNOT CONTINUE TESTING IF NOT
8426  027556  000000                            HALT                    ;STOP TEST
8427  027560  013711  002344                    MOV     @#WRIFOR,@R1    ;GET READY FOR WRITE HEADER AND
8428                                                                    ;DATA WITH 62 IN RHCS1
8429  027564  005037  004632                    CLR     @#ERFLG$        ;CLEAR ERROR FLAG
8430  027570  012777  010000  152412            MOV     #FMT22,@RHOF    ;FORMAT BIT=1 (16 BIT WORDS)
8431  027576  005077  152410                    CLR     @RHCA           ;CYLINDER =0
8432
8433                                            ;*SAVE REGISTERS FOR COMPARISON AFTER READ
8434  027602  004037  033462                    JSR     R0,@#SAVER      ;SAVE REGISTERS
8435  027606  002172                            RHWC                    ;RHWC IS THE FIRST REGISTER SAVED
8436  027610  004512                            SAVERE                  ;STARTING ADDRES OF WHERE
8437                                                                    ;THE REGISTERS ARE SAVED
8438  027612  000023                            19.                     ;NUMBER OF REGISTERS
8439                                                                    ;SAVED = 19.
```

```
8440
8441                                        ;*GO TO WRITE HEADER AND DATA
8442                                        ;*BUT BEFORE GO, ONE INDEX PULSE IS GIVEN
8443                                        ;*TO CLEAR OUT THE SECTOR CLOCK COUNTER IN THE RH11
8444                                        ;*SO THAT NO SECTOR PULSES COME DURING THIS TEST
8445
8446  027614  013700  002220                MOV     @#RHMR,R0       ;NOW R0 HAS MAINTENANCE REG. ADDR.
8447  027620  012710  000001                MOV     #DMD,@R0        ;SET DIAGNOSTIC MODE
8448  027624  052710  000004                BIS     #MINX,@R0       ;SET INDEX
8449  027630  042710  000004                BIC     #MINX,@R0       ;CLEAR INDEX THIS GIVES
8450                                                                ;ONE INDEX PULSE
8451
8452  027634  052777  000001  152336        BIS     #GO,@RHCS1      ;ISSUE THE 'GO' BIT TO THE RH11
8453  027642  012737  000113  004642  RUNWAT: MOV     #75.,@#RUNCTR   ;LOAD 'RUN' LINE DELAY COUNTER
8454                                                                ;= APPROX 450 US ON 11/50 CPU WITH CORE
8455                                                                ;AND PROVIDES FOR TIME TO FILL THE SILO
8456  027650  005337  004642          1$:     DEC     @#RUNCTR        ;COUNT DOWN ONCE
8457  027654  001375                        BNE     1$              ;CONTINUE UNTIL = 0
8458
8459                                        ;*ISSUE THE FIRST DIAGNOSTIC INDEX PULSE
8460  027656  052710  000004                BIS     #MINX,@R0       ;SET INDEX PULSE
8461  027662  042710  000004                BIC     #MINX,@R0       ;RESET INDEX
8462
8463                                        ;*SECOND INDEX PULSE
8464  027666  052710  000004                BIS     #MINX,@R0       ;SET INDEX
8465  027672  042710  000004                BIC     #MINX,@R0       ;CLEAR INDEX
8466
8467                                        ;*THIRD INDEX PULSE
8468  027676  052710  000004                BIS     #MINX,@R0       ;SET INDEX
8469  027702  042710  000004                BIC     #MINX,@R0       ;CLEAR INDEX
8470
8471
8472                                        ;*CHANGE SAVED REGISTERS TO EXPECTED VALUE
8473
8474  027706  004037  034204                JSR     R0,@#CHREG      ;CHANGE BITS IN SAVED REGISTER
8475  027712  002200                        RHCS1                   ;CHANGE RHCS1 REGISTER
8476
8477  027714  000002                        2                       ;2 BIT/BITS TO BE CHANGED
8478  027716  000001                        1                       ;NEW VALUE OF SC IS 1
8479  027720  100000                        SC                      ;CHANGE SC BIT
8480  027722  000001                        1                       ;NEW VALUE OF TRE IS 1
8481  027724  040000                        TRE                     ;CHANGE TRE BIT
8482
8483  027726  004037  034204                JSR     R0,@#CHREG      ;CHANGE BITS IN SAVED REGISTER
8484  027732  002222                        RHDS1                   ;CHANGE RHDS1 REGISTER
8485
8486  027734  000002                        2                       ;2 BIT/BITS TO BE CHANGED
8487  027736  000001                        1                       ;NEW VALUE OF ATA IS 1
8488  027740  100000                        ATA                     ;CHANGE ATA BIT
8489  027742  000001                        1                       ;NEW VALUE OF ERR IS 1
8490  027744  040000                        ERR                     ;CHANGE ERR BIT
8491
8492  027746  004037  033216                JSR     R0,@#FILLRE     ;MOV 2 INTO SAVED RHDST
8493  027752  002204                        RHDST                   ;SAVED REGISTER TO CHANGE
8494  027754  000002                        2                       ;DATA
8495
```

```
 8496
 8497  027756  004037  034204                JSR     R0,@#CHREG      ;CHANGE BITS IN SAVED REGISTER
 8498  027762  002202                        RHER1                   ;CHANGE RHER1 REGISTER
 8499
 8500  027764  000001                        1                       ;1 BIT/BITS TO BE CHANGED
 8501  027766  000001                        1                       ;NEW VALUE OF OPI IS 1
 8502  027770  020000                        OPI                     ;CHANGE OPI BIT
 8503
 8504  027772  053737  004644  004536        BIS     @#ATTENT,@#SAVERE+24 ;SET APPROPIATE 'ATA' BITS
 8505                                                                ;FOR WORKING DRIVE IN
 8506                                                                ;SAVED RHAS LOCATION
 8507
 8508  030000  004037  034204                JSR     R0,@#CHREG      ;CHANGE BITS IN SAVED REGISTER
 8509  030004  002220                        RHMR                    ;CHANGE RHMR REGISTER
 8510
 8511  030006  000001                        1                       ;1 BIT/BITS TO BE CHANGED
 8512  030010  000001                        1                       ;NEW VALUE OF DMD IS 1
 8513  030012  000001                        DMD                     ;CHANGE DMD BIT
 8514
 8515
 8516                                        ;*RHWC,RHBA AND OR AND IR BITS OF RHCS2 WILL NOT BE CHECKED
 8517  030014  017737  152152  004512        MOV     @RHWC,@#SAVERE  ;SAVED RHWC
 8518  030022  017737  152146  004514        MOV     @RHBA,@#SAVERE+2;SAVED RHBA
 8519  030030  017746  152142                MOV     @RHCS2,-(SP)    ;GET RHCS2
 8520  030034  042716  177477                BIC     #^C<IR!OR>,(SP) ;GET 'IR' & 'OR' STATES
 8521  030040  042737  000300  004516        BIC     #IR!OR,@#SAVERE+4;CLEAR 'IR' & 'OR' BITS
 8522  030046  052637  004516                BIS     (SP)+,@#SAVERE+4;SET 'OR' & 'IR' AS REQUIRED
 8523
 8524
 8525                                        ;*COMPARE REGISTERS BEFORE WRITE WITH RESULTS AFTER WRITE
 8526
 8527  030052  004037  034312                JSR     R0,@#COMREG     ;COMPARE SAVED REGISTERS WITH
 8528                                                                ;PRESENT VALUE
 8529  030056  004512                        SAVERE                  ;GOOD DATA SAVED IN 'SAVERE'
 8530  030060  002254                        WC                      ;TEST DATA STARTING FROM 'RHWC'
 8531  030062  000021                        17.                     ;17. REGISTERS TO BE COMPARED
 8532  030064  030070                        2$                      ;RETURN TO 2$ ON ERROR
 8533  030066  030074                        3$                      ;RETURN TO 3$ ON NO ERROR
 8534
 8535
 8536  030070  104071          2$:           ERROR   71              ;FORCING OPI CAUSED
 8537  030072  000207                        RTS     PC              ;IMPROPER REGISTER CHANGE
 8538                                                                ;GOOD DATA GIVES WHAT SHOULD BE THERE
 8539                                                                ;RECEIVED DATA GIVES WHAT WAS THERE
 8540                                                                ;AFTER 3 INDEX PULSES WERE ISSUED
 8541
 8542
 8543  030074  004737  033314          3$:   JSR     PC,@#CLDISK     ;CLEAR THE 'GO' BIT
```

```
8544
8545                                    ;;*****************************************************************
8546                                    ;*TEST 34        CONSECUTIVE SECTOR FORMATTING
8547
8548                                    ;*      46 CONSECUTIVE SECTORS WILL BE FORMATTED
8549                                    ;*      STARTING FROM CYLINDER 0 TRACK 0 SECTOR 21.
8550                                    ;*      FORMATTING WILL BE DONE BY A WRITE HEADER AND
8551                                    ;*      DATA COMMAND FOR 4 WORDS, ONE SECTOR
8552                                    ;*      AT A TIME.
8553                                    ;*
8554                                    ;*      AFTER EACH SECTOR IS WRITTEN, 'SC' WILL BE CHECKED
8555                                    ;*      TO INSURE THAT THERE WERE NO ERRORS
8556
8557                                    ;;*****************************************************************
8558  030100  000004                    TST34:  SCOPE
8559  030102  012706  001000                    MOV     #STACK,SP       ;RESET STACK
8560  030106  012737  000034  004504            MOV     #34,@#TSTNM     ;SAVE TEST NUMBER
8561
8562  030114  004737  033314                    JSR     PC,@#CLDISK     ;SET R1-RHCS1, R2-RHCS2
8563                                                                    ;R3-RHDS1, R4-RHER1
8564                                                                    ;GIVE RH-11 INITIALIZE
8565                                                                    ;SETUP UNIT NUMBER
8566  030120  012737  000025  030154            MOV     #21.,@#1$+12    ;SET UP TO START FROM
8567  030126  012737  000025  030170            MOV     #21.,@#2$+6     ;SECTOR 21.
8568  030134  012737  000056  004654            MOV     #46.,@#TMP1     ;46 SECTORS TO COVER 3 TRACKS
8569
8570                                            ;*FILL WRITE FROM BUFFER WITH THE HEADER
8571  030142                            1$:
8572
8573  030142  004037  033140                    JSR     R0,@#FLHEAD     ;SAVE HEADER DATA IN WRFROM
8574  030146  002370                            WRFROM                  ;LOCATION WHERE SAVED
8575  030150  000004                            4                       ;NUMBER OF WORDS SAVED
8576  030152  010000                            10000                   ;FIRST DATA WORD
8577  030154  000025                            21.                     ;SECOND DATA WORD
8578  030156  000000                            0                       ;THIRD DATA WORD
8579  030160  000000                            0                       ;FOURTH DATA WORD
8580
8581                                            ;*NOW THE WRITE HEADER AND DATA COMMAND WILL BE SETUP
8582  030162                            2$:
8583
8584  030162  004037  035276                    JSR     R0,@#RUN        ;SETUP TO RUN FOR DATA COMMAND
8585  030166  000000                            0                       ;CYLINDER 0
8586  030170     025                            .BYTE   21.             ;SECTOR 21.
8587  030171     000                            .BYTE   0               ;TRACK 0
8588  030172  177774                            -0-4                    ;WORD COUNT (DATA) = 0 +
8589                                                                    ;4 HEADER WORDS
8590  030174  002370                            WRFROM                  ;BUS ADDRESS
8591                                                                    ;STARTING ADDRESS OF DATA
8592                                                                    ;BUFFER = WRFROM
8593  030176  000000                            0                       ;DO NOT INHIBIT BUS ADDRESS INCREMENT
8594  030200  010000                            FMT22                   ;16 BITS PER WORD FORMAT
8595                                                                    ;DO NOT INHIBIT ECC CORRECTION
8596                                                                    ;DO NOT INHIBIT HEADER COMPARE
8597  030202  002344                            WRIFOR                  ;GET READY TO DO A WRIFOR
8598                                                            ;WRITE HEADER AND DATA WITH 62 IN RHCS1
8599
```

```
8600
8601  030204  004737  033374              JSR     PC,@#CHECKT     ;CHECKS DVA,RDY,MOL,DPR,DRY AND VV = 1
8602                                                              ;AND THAT ALL STATUS BITS ARE = 0
8603  030210  104401  057465              TYPE    ,CPHALT         ;CANNOT CONTINUE TESTING IF NOT
8604  030214  000000                      HALT                    ;STOP TEST
8605
8606  030216  013777  004506  151742      MOV     @#RP4VEC,@RPVEC ;SET RP04 VECTOR ADDRESS
8607                                                              ;TO 'TIME1' IF P-CLOCK IS PRESENT
8608                                                              ;OR TO 'TIME2' IF P-CLOCK IS NOT THERE
8609                                                              ;'TIME' WILL ONLY SAVE
8610                                                              ;CURRENT CYLINDER ADDRESS
8611                                                              ;AND LOOK AHEAD REGISTERS
8612
8613
8614  030224  013746  002344              MOV     @#WRIFGR,-(SP)  ;GET READY TO MOVE COMMAND
8615  030230  052716  000101              BIS     #GO!IE,(SP)     ;GET READY TO SET GO AND
8616                                                              ;ENABLE INTERRUPT
8617  030234  012677  151740              MOV     (SP)+,@RHCS1    ;GO WITH
8618                                                               ;62 IN RHCS1 FOR WRITE HEADER AND DATA
8619                                                              ;WITH INTERRUPT ENABLED
8620  030240  011100                      MOV     @R1,R0          ;SAVE RHCS1 DURING ABOVE OPERATION
8621  030242  011305                      MOV     @R3,R5          ;SAVE RHDS1 DURING ABOVE OPERATION
8622
8623                                      ;*ONE REVOLUTION=16670 MICRO SEC, ONE SECTOR=760 MICRO SEC
8624
8625  030244  104413                      WAT                     ;WAIT FOR RDY BIT TO SET
8626  030246  002200                      RHCS1                   ;WAIT FOR RHCS1 REGISTER
8627  030250  000200                      RDY                     ;WAIT FOR RDY BIT IN RHCS1 REGISTER
8628  030252  003237                      1695.                   ;ALLOW 16950 MICRO SECONDS
8629  030254  001515                      845.                    ;RDY MUST SET BETWEEN
8630                                                              ;8500 AND 25400 MICRO SECONDS
8631
8632                                      ;*NOW ONE MORE SECTOR HAS BEEN WRITTEN
8633                                      ;*'SC' WILL BE CHECKED TO MAKE SURE
8634                                      ;*NO ERRORS OCCURED
8635
8636  030256  017737  151716  002262      MOV     @RHCS1,@#CS1    ;GET RHCS1
8637  030264  032737  100000  002262      BIT     #SC,@#CS1       ;IS 'SC' SET ?
8638  030272  001403                      BEQ     3$              ;BRANCH IF ''SPECIAL CONDITION'' NOT SET
8639  030274  004737  035236              JSR     PC,@#PUTREG     ;READ & SAVE  ALL RH11 & RP04 REGISTERS
8640  030300  104072                      ERROR   72              ;THERE WAS AN UNDEFINED ERROR AFTER
8641                                                              ;A WRITE HEADER AND DATA
8642
8643                                      ;*A SECTOR HAS BEEN FORMATTED NOW,
8644                                      ;*THE HARDWARE WILL BE CLEARED AND
8645                                      ;*CHANGES WILL BE MADE TO FORMAT NEXT SECTOR.
8646
8647  030302                      3$:
8648
8649  030302  004737  033314              JSR     PC,@#CLDISK     ;SET R1-RHCS1, R2-RHCS2
8650                                                              ;R3-RHDS1, R4-RHER1
8651                                                              ;GIVE RH-11 INITIALIZE
8652                                                              ;SETUP UNIT NUMBER
8653  030306  013705  030154              MOV     @#1$+12,R5      ;GET SECTOR TRACK WORD
8654  030312  005205                      INC     R5              ;+ 1
8655  030314  122705  000026              CMPB    #22.,R5         ;IS IT 22 SECTORS (WHOLE TRACK DONE) ?
```

```
 8656  030320  001405                        BEQ     4$                ;YES...DO NEXT TRACK
 8657  030322  010537  030154                MOV     R5,@#1$+12        ;NO...RESTORE SECTOR TRACK FOR DATA
 8658  030326  010537  030170                MOV     R5,@#2$+6         ;RESTORE SECTOR TRACK FOR "RUN" ROUTINE
 8659  030332  000410                        BR      5$                ;CHECK FOR 46 SECTORS COMPLETED
 8660
 8661  030334  105037  030154          4$:   CLRB    @#1$+12           ;SET SECTOR = 0 FOR DATA WRITTEN
 8662  030340  105237  030155                INCB    @#1$+13           ;INCR TRACK FOR DATA WRITTEN
 8663  030344  105037  030170                CLRB    @#2$+6            ;SET SECTOR = 0 FOR "RUN" ROUTINE
 8664  030350  105237  030171                INCB    @#2$+7            ;INCR TRACK FOR THE "RUN" ROUTINE
 8665
 8666  030354  005337  004654          5$:   DEC     @#TMP1            ;ARE 46 SECTORS DONE ?
 8667  030360  001270                        BNE     1$                ;CONTINUE FORMATTING IF NOT
 8668
 8669  030362                           6$:                            ;GO ON TO NEXT TEST IF SO
```

```
                                ;;*****************************************************************
                                ;*TEST 35       OPERATION INCOMPLETE - RHER1 (BIT #13)OPI

                                ;*      THIS WILL TEST THAT OPI DOES NOT SET WHEN THREE NORMAL
                                ;*      INDEX PULSES ARE ENCOUNTERED IN A READ COMMAND

                                ;*      FIRST 46 CONSECUTIVE SECTORS WILL BE FORMATTED
                                ;*      STARTING FROM CYLINDER 0 TRACK 0 SECTOR 21.
                                ;*      FORMATTING WILL BE DONE BY A WRITE HEADER AND
                                ;*      DATA COMMAND FOR 4 WORDS, ONE SECTOR
                                ;*      AT A TIME
                                ;*
                                ;*      THEN A READ HEADER AND DATA WILL BE DONE
                                ;*      FOR CYLINDER 0 TRACK 0 SECTOR 21 FOR
                                ;*      11960 WORDS (260.X22X2+260+260) WITH BUS
                                ;*      ADDRESS INHIBIT SET.
                                ;*
                                ;*      AT THE END ALL REGISTERS WILL BE CHECKED.

                                ;;*****************************************************************
030362  000004                  TST35:  SCOPE
030364  012706  001000                  MOV     #STACK,SP       ;RESET STACK
030370  012737  000035  004504          MOV     #35,@#TSTNM     ;SAVE TEST NUMBER

030376  004737  033314                  JSR     PC,@#CLDISK     ;SET R1-RHCS1, R2-RHCS2
                                                                ;R3-RHDS1, R4-RHER1
                                                                ;GIVE RH-11 INITIALIZE
                                                                ;SETUP UNIT NUMBER
030402  012737  000025  030436          MOV     #21.,@#1$+12    ;SET UP TO START FROM
030410  012737  000025  030452          MOV     #21.,@#2$+6     ;SECTOR 21.
030416  012737  000056  004654          MOV     #46.,@#TMP1     ;46 SECTORS TO COVER 3 TRACKS

                                        ;*FILL WRITE FROM BUFFER WITH HEADER
030424                          1$:

030424  004037  033140                  JSR     R0,@#FLHEAD     ;SAVE HEADER DATA IN WRFROM
030430  002370                          WRFROM                  ;LOCATION WHERE SAVED
030432  000004                          4                       ;NUMBER OF WORDS SAVED
030434  010000                          10000                   ;FIRST DATA WORD
030436  000025                          21.                     ;SECOND DATA WORD
030440  000000                          0                       ;THIRD DATA WORD
030442  000000                          0                       ;FOURTH DATA WORD

                                        ;*NOW THE WRITE HEADER AND DATA COMMAND WILL BE FILLED
030444                          2$:

030444  004037  035276                  JSR     R0,@#RUN        ;SETUP TO RUN FOR DATA COMMAND
030450  000000                          0                       ;CYLINDER 0
030452     025                  .BYTE   21.                     ;SECTOR 21.
030453     000                  .BYTE   0                       ;TRACK 0
030454  177774                          -0-4                    ;WORD COUNT (DATA) = 0 +
                                                                ;4 HEADER WORDS
030456  002370                          WRFROM                  ;BUS ADDRESS
                                                                ;STARTING ADDRESS OF DATA
```

```
 8726
 8727  030460  000000                          0                       ;BUFFER = WRFROM
 8728  030462  010000                          FMT22                   ;16 BITS PER WORD FORMAT
 8729                                                                  ;DO NOT INHIBIT ECC CORRECTION
 8730                                                                  ;DO NOT INHIBIT HEADER COMPARE
 8731  030464  002344                          WRIFOR                  ;GET READY TO DO A WRIFOR
 8732                                                        ;WRITE HEADER AND DATA WITH 62 IN RHCS1
 8733
 8734
 8735  030466  004737  033374                  JSR     PC,@#CHECKT     ;CHECKS DVA,RDY,MOL,DPR,DRY AND VV = 1
 8736                                                                  ;AND THAT ALL STATUS BITS ARE = 0
 8737  030472  104401  057465                  TYPE    ,CPHALT         ;CANNOT CONTINUE TESTING IF NOT
 8738  030476  000000                          HALT                    ;STOP TEST
 8739
 8740  030500  013777  004506  151460          MOV     @#RP4VEC,@RPVEC ;SET RP04 VECTOR ADDRESS
 8741                                                                  ;TO 'TIME1' IF P-CLOCK IS PRESENT
 8742                                                                  ;OR TO 'TIME2' IF P-CLOCK IS NOT THERE
 8743                                                                  ;'TIME' WILL ONLY SAVE
 8744                                                                  ;CURRENT CYLINDER ADDRESS
 8745                                                                  ;AND LOOK AHEAD REGISTERS
 8746
 8747
 8748  030506  013746  002344                  MOV     @#WRIFOR,-(SP)  ;GET READY TO MOVE COMMAND
 8749  030512  052716  000101                  BIS     #GO!IE,(SP)     ;GET READY TO SET GO AND
 8750                                                                  ;ENABLE INTERRUPT
 8751  030516  012677  151456                  MOV     (SP)+,@RHCS1    ;GO WITH
 8752                                                                   ;62 IN RHCS1 FOR WRITE HEADER AND DATA
 8753                                                                  ;WITH INTERRUPT ENABLED
 8754  030522  011100                          MOV     @R1,R0          ;SAVE RHCS1 DURING ABOVE OPERATION
 8755  030524  011305                          MOV     @R3,R5          ;SAVE RHDS1 DURING ABOVE OPERATION
 8756
 8757                                          ;*ONE REVOLUTION=16670 MICRO SEC, ONE SECTOR=760 MICRO SEC
 8758
 8759  030526  104413                          WAT                     ;WAIT FOR RDY BIT TO SET
 8760  030530  002200                          RHCS1                   ;WAIT FOR RHCS1 REGISTER
 8761  030532  000200                          RDY                     ;WAIT FOR RDY BIT IN RHCS1 REGISTER
 8762  030534  003237                          1695.                   ;ALLOW 16950 MICRO SECONDS
 8763  030536  001515                          845.                    ;RDY MUST SET BETWEEN
 8764                                                                  ;8500 AND 25400 MICRO SECONDS
 8765
 8766                                          ;*NOW ONE MORE SECTOR HAS BEEN WRITTEN
 8767                                          ;*'SC' WILL BE CHECKED TO MAKE SURE
 8768                                          ;*NO ERRORS OCCURED
 8769
 8770  030540  017737  151434  002262          MOV     @RHCS1,@#CS1    ;GET RHCS1
 8771  030546  032737  100000  002262          BIT     #SC,@#CS1       ;IS 'SC' SET ?
 8772  030554  001405                          BEQ     3$              ;BRANCH IF "SPECIAL CONDITION" NOT SET
 8773  030556  004737  035236                  JSR     PC,@#PUTREG     ;READ & SAVE  ALL RH11 & RP04 REGISTERS
 8774  030562  104072                          ERROR   72              ;THERE WAS AN UNDEFINED ERROR AFTER
 8775                                                                  ;A WRITE HEADER AND DATA
 8776
 8777  030564  000137  031020                  JMP     TST36           ;       THIS IS A SETUP ERROR AND 'OPI' TEST CAN'T CONTI
 8778                                                                  ;GO ON TO NEXT TEST
 8779
 8780                                          ;*ONE SECTOR HAS BEEN FORMATTED NOW,
 8781                                          ;*THE HARDWARE WILL BE CLEARED AND
```

```
 8782                                        ;*CHANGES WILL BE MADE TO FORMAT NEXT SECTOR.
 8783
 8784  030570                          3$:
 8785
 8786  030570  004737  033314              JSR     PC,@#CLDISK     ;SET R1-RHCS1, R2-RHCS2
 8787                                                              ;R3-RHDS1, R4-RHER1
 8788                                                              ;GIVE RH-11 INITIALIZE
 8789                                                              ;SETUP UNIT NUMBER
 8790  030574  013705  030436              MOV     @#1$+12,R5      ;GET SECTOR TRACK WORD
 8791  030600  005205                      INC     R5              ;+ 1
 8792  030602  122705  000026              CMPB    #22.,R5         ;IS IT 22 (WHOLE TRACK) ?
 8793  030606  001405                      BEQ     4$              ;YES...DO NEXT TRACK
 8794  030610  010537  030436              MOV     R5,@#1$+12      ;NO...RESTORE SECTOR TRACK FOR DATA WRITTEN
 8795  030614  010537  030452              MOV     R5,@#2$+6       ;RESTORE SECTOR TRACK FOR "RUN" ROUTINE
 8796  030620  000410                      BR      5$              ;CHECK FOR 46 SECTORS COMPLETED
 8797
 8798  030622  105037  030436        4$:   CLRB    @#1$+12         ;SET SECTOR = 0 FOR DATA WRITTEN
 8799  030626  105237  030437              INCB    @#1$+13         ;INCR TRACK FOR THE "RUN" ROUTINE
 8800  030632  105037  030452              CLRB    @#2$+6          ;SET SECTOR = 0 FOR DATA WRITTEN
 8801  030636  105237  030453              INCB    @#2$+7          ;INCR TRACK FOR THE "RUN" ROUTINE
 8802
 8803  030642  005337  004654        5$:   DEC     @#TMP1          ;ARE 46 SECTORS DONE ?
 8804  030646  001266                      BNE     1$              ;CONTINUE IF NOT
 8805
 8806
 8807                                        ;*NOW 46 SECTORS HAVE BEEN FORMATTED
 8808
 8809                                        ;*READ HEADER AND DATA FOR 46 SECTORS=11960 WORDS
 8810                                        ;*WITH BUS ADDRESS INHIBITED
 8811
 8812
 8813  030650  004737  033314              JSR     PC,@#CLDISK     ;SET R1-RHCS1, R2-RHCS2
 8814                                                              ;R3-RHDS1, R4-RHER1
 8815                                                              ;GIVE RH-11 INITIALIZE
 8816                                                              ;SETUP UNIT NUMBER
 8817
 8818                                        ;*FILL READ HEADER AND DATA COMMAND
 8819
 8820
 8821  030654  004037  035276              JSR     R0,@#RUN        ;SETUP TO RUN FOR DATA COMMAND
 8822  030660  000000                      0                       ;CYLINDER 0
 8823  030662     025                      .BYTE   21.             ;SECTOR 21.
 8824  030663     000                      .BYTE   0               ;TRACK 0
 8825  030664  150510                      -11956.-4                       ;WORD COUNT (DATA) = 11956. +
 8826                                                              ;4 HEADER WORDS
 8827  030666  003434                      REINTO                  ;BUS ADDRESS
 8828                                                              ;STARTING ADDRESS OF DATA
 8829                                                              ;BUFFER = REINTO
 8830  030670  000010                      BAI                     ;INHIBIT BUS ADDRESS INCREMENT
 8831  030672  014000                      FMT22!ECI                       ;16 BITS PER WORD FORMAT
 8832                                                              ;INHIBIT ECC CORRECTION
 8833                                                              ;DO NOT INHIBIT HEADER COMPARE
 8834  030674  002350                      REFOR                   ;GET READY TO DO A REFOR
 8835                                                              ;READ HEADER AND DATA WITH 72 IN RHCS1
 8836
 8837  030676  004737  033374              JSR     PC,@#CHECKT     ;CHECKS DVA,RDY,MOL,DPR,DRY AND VV = 1
```

```
                                                        ;AND THAT ALL STATUS BITS ARE = 0
030702  104401  057465                TYPE    ,CPHALT   ;CANNOT CONTINUE TESTING IF NOT
030706  000000                        HALT              ;STOP TEST
030710  013777  004506  151250        MOV     @#RP4VEC,@RPVEC ;SET RP04 VECTOR ADDRESS
                                                        ;TO 'TIME1' IF P-CLOCK IS PRESENT
                                                        ;OR TO 'TIME2' IF P-CLOCK IS NOT THERE
                                                        ;'TIME' WILL ONLY SAVE
                                                        ;CURRENT CYLINDER ADDRESS
                                                        ;AND LOOK AHEAD REGISTERS

030716  013746  002350                MOV     @#REFOR,-(SP) ;GET READY TO MOVE COMMAND
030722  052716  000101                BIS     #GO!IE,(SP)   ;GET READY TO SET GO AND
                                                        ;ENABLE INTERRUPT
030726  012677  151246                MOV     (SP)+,@RHCS1  ;GO WITH
                                                          ;72 IN RHCS1 FOR READ DATA
                                                        ;WITH INTERRUPT ENABLED
030732  011100                        MOV     @R1,R0        ;SAVE RHCS1 DURING ABOVE OPERATION
030734  011305                        MOV     @R3,R5        ;SAVE RHDS1 DURING ABOVE OPERATION

                                      ;*TIME IS NOT IMPORTANT

030736  104413                        WAT                   ;WAIT FOR RDY BIT TO SET
030740  002200                        RHCS1                 ;WAIT FOR RHCS1 REGISTER
030742  000200                        RDY                   ;WAIT FOR RDY BIT IN RHCS1 REGISTER
030744  121320                        41680.                ;ALLOW 416800 MICRO SECONDS
030746  121320                        41680.                ;RDY MUST SET BETWEEN
                                                        ;00 AND 178240 MICRO SECONDS

                                      ;*NOW THAT ALL 11960 WORDS HAVE BEEN READ
                                      ;*'OPI' WILL BE CHECKED TO BE NOT SET

030750  017737  151226  002264        MOV     @RHER1,@#ER1  ;GET RHER1
030756  032737  020000  002264        BIT     #OPI,@#ER1    ;IS 'OPI' SET ?
030764  001403                        BEQ     6$            ;CHECK 'SC' IF NOT
030766  004737  035236                JSR     PC,@#PUTREG   ;READ & SAVE ALL RH11 & RP04 REGISTERS
030772  104074                        ERROR   74            ;READ HEADER AND DATA
                                                        ;OVER 3 INDEX PULSES
                                                        ;CAUSED 'OPI' TO SET

                                      ;*'SC' WILL BE CHECKED

030774  017737  151200  002262  6$:   MOV     @RHCS1,@#CS1  ;GET RHCS1
031002  032737  100000  002262        BIT     #SC,@#CS1     ;IS 'SC' SET ?
031010  001403                        BEQ     7$            ;CONTINUE TESTING IF NOT
031012  004737  035236                JSR     PC,@#PUTREG   ;READ & SAVE ALL RH11 & RP04 REGISTERS
031016  104072                        ERROR   72            ;READ HEADER AND DATA
                                                        ;FOR 11960 WORDS, THAT IS OVER THREE
                                                        ;INDEX PULSES, CAUSED AN UNDEFINED ERROR

031020                          7$:                   ;CONTINUE WITH THE NEXT TEST
```

```
 8889
 8890
 8891                                   ;;*****************************************************************
 8892                                   ;*TEST 36       HEAD SELECTION TEST ERR & TRE
 8893                                   ;*      THIS TESTS HEAD SELECTION LOGIC ONLY. A WRITE HEADER AND
 8894                                   ;*      DATA COMMAND IS GIVEN TO EACH TRACK FROM 0 TO 18 ON
 8895                                   ;*      CYLINDER 0, SECTOR 0.
 8896                                   ;*      THE DATA ON EACH SECTOR IS UNIQUE. THE LEAST SIGNIFICANT
 8897                                   ;*      5 BITS GIVE SECTOR THE NEXT LEAST SIGNIFICANT 5 BITS
 8898                                   ;*      GIVE TRACK THE NEXT 6 BITS GIVE CYLINDER
 8899                                   ;*
 8900                                   ;*      THEN READ HEADER AND DATA IS DONE FOR THE ABOVE AND DATA
 8901                                   ;*      CHECKED
 8902                                   ;*
 8903                                   ;*      BETWEEN THE WRITE AND READ ONLY "ERR" AND "TRE" ARE CHECKED
 8904                                   ;*
 8905                                   ;*
 8906                                   ;*      ON AN ERROR IN THE READ HEADER AND DATA LOOPING WILL BE
 8907                                   ;*      ONLY ON THE ERROR SECTOR READ
 8908
 8909                                   ;;*****************************************************************
 8910  031020  000004                  TST36:  SCOPE
 8911  031022  012706  001000                  MOV     #STACK,SP       ;RESET STACK
 8912  031026  012737  000036  004504          MOV     #36,@#TSTNM     ;SAVE TEST NUMBER
 8913
 8914  031034  004737  033314                  JSR     PC,@#CLDISK     ;SET R1-RHCS1, R2-RHCS2
 8915                                                                   ;R3-RHDS1, R4-RHER1
 8916                                                                   ;GIVE RH-11 INITIALIZE
 8917                                                                   ;SETUP UNIT NUMBER
 8918
 8919                                  ;*THE FOLLOWING CLEARS ARE TO INITIALIZE TEST FROM CYLINDER 0
 8920  031040  005037  031146          CLR     @#1$+12         ;START WITH SECTOR/TRACK = 0
 8921  031044  005037  031164          CLR     @#2$+10         ;START WITH DATA = 0
 8922  031050  005037  031174          CLR     @#3$+6          ;START WITH 0 FOR COMMAND
 8923
 8924  031054  012737  000023  001200  MOV     #19.,@#$TMP1    ;19 TRACKS TO BE WRITTEN
 8925
 8926                                  ;*THIS GETS THE HEADS TO CYLINDER 0
 8927
 8928  031062  004737  033374          JSR     PC,@#CHECKT     ;CHECKS DVA,RDY,MOL,DPR,DRY AND VV = 1
 8929                                                           ;AND THAT ALL STATUS BITS ARE = 0
 8930  031066  104401  057465          TYPE    ,CPHALT         ;CANNOT CONTINUE TESTING IF NOT
 8931  031072  000000                  HALT                    ;STOP TEST
 8932
 8933  031074  013777  004506  151064  MOV     @#RP4VEC,@RPVEC ;SET RP04 VECTOR ADDRESS
 8934                                                           ;TO 'TIME1' IF P-CLOCK IS PRESENT
 8935                                                           ;OR TO 'TIME2' IF P-CLOCK IS NOT THERE
 8936                                                           ;'TIME' WILL ONLY SAVE
 8937                                                           ;CURRENT CYLINDER ADDRESS
 8938                                                           ;AND LOOK AHEAD REGISTERS
 8939
 8940
 8941  031102  013746  002326          MOV     @#RECAL1,-(SP)  ;GET READY TO MOVE COMMAND
 8942  031106  052716  000101          BIS     #GO!IE,(SP)     ;GET READY TO SET GO AND
 8943                                                           ;ENABLE INTERRUPT
 8944  031112  012677  151062          MOV     (SP)+,@RHCS1    ;GO WITH
```

```
                                                        ;6 IN RHCS1 FOR RECALIBRATE
                                                       ;WITH INTERRUPT ENABLED

031116  104413                        WAT             ;WAIT FOR DRY BIT TO SET
031120  002222                        RHDS1           ;WAIT FOR RHDS1 REGISTER
031122  000200                        DRY             ;WAIT FOR DRY BIT IN RHDS1 REGISTER
031124  060650                        25000.          ;ALLOW 250000 MICRO SECONDS
031126  060650                        25000.          ;DRY MUST SET BETWEEN
                                                       ;00 AND 500000 MICRO SECONDS

031130  004737  033314                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
031134                          1$:
031134  004037  033140                JSR     R0,@#FLHEAD     ;SAVE HEADER DATA IN WRFROM
031140  002370                        WRFROM          ;LOCATION WHERE SAVED
031142  000004                        4               ;NUMBER OF WORDS SAVED
031144  010000                        10000           ;FIRST DATA WORD
031146  000000                        <0*400>!0               ;SECOND DATA WORD
031150  000000                        0               ;THIRD DATA WORD
031152  000000                        0               ;FOURTH DATA WORD

                                    ;*FILL WRITE FROM BUFFER WITH DATA
031154                          2$:
031154  004037  033164                JSR     R0,@#CLAREA     ;CLEAR 256. WORDS, FROM WRFROM+10
031160  002400                        WRFROM+10               ;STARTING FROM WRFROM+10
031162  000400                        256.            ;256. WORDS
031164  000000                        <0.*2000>!<0.*40>!0             ;FILL WITH <0.*2000>!<0.*40>!0

                                    ;*THE WRITE HEADER AND DATA COMMAND WILL BE FILLED
031166                          3$:
031166  004037  035276                JSR     R0,@#RUN        ;SETUP TO RUN FOR DATA COMMAND
031172  000000                        0               ;CYLINDER 0
031174     000                      .BYTE   0               ;SECTOR 0
031175     000                      .BYTE   0               ;TRACK 0
031176  177374                      -256.-4                 ;WORD COUNT (DATA) = 256. +
                                                       ;4 HEADER WORDS
031200  002370                        WRFROM          ;BUS ADDRESS
                                                       ;STARTING ADDRESS OF DATA
                                                       ;BUFFER = WRFROM
031202  000000                        0               ;DO NOT INHIBIT BUS ADDRESS INCREMENT
031204  010000                        FMT22           ;16 BITS PER WORD FORMAT
                                                       ;DO NOT INHIBIT ECC CORRECTION
                                                       ;DO NOT INHIBIT HEADER COMPARE
031206  002344                        WRIFOR          ;GET READY TO DO A WRIFOR
                                                    ;WRITE HEADER AND DATA WITH 62 IN RHCS1
```

```
9001
9002
9003  031210  004737  033374              JSR     PC,@#CHECKT     ;CHECKS DVA,RDY,MOL,DPR,DRY AND VV = 1
9004                                                              ;AND THAT ALL STATUS BITS ARE = 0
9005  031214  104401  057465              TYPE    ,CPHALT         ;CANNOT CONTINUE TESTING IF NOT
9006  031220  000000                      HALT                    ;STOP TEST
9007
9008  031222  013777  004506  150736      MOV     @#RP4VEC,@RPVEC ;SET RP04 VECTOR ADDRESS
9009                                                              ;TO 'TIME1' IF P-CLOCK IS PRESENT
9010                                                              ;OR TO 'TIME2' IF P-CLOCK IS NOT THERE
9011                                                              ;'TIME' WILL ONLY SAVE
9012                                                              ;CURRENT CYLINDER ADDRESS
9013                                                              ;AND LOOK AHEAD REGISTERS
9014
9015
9016  031230  013746  002344              MOV     @#WRIFOR,-(SP)  ;GET READY TO MOVE COMMAND
9017  031234  052716  000101              BIS     #GO!IE,(SP)     ;GET READY TO SET GO AND
9018                                                              ;ENABLE INTERRUPT
9019  031240  012677  150734              MOV     (SP)+,@RHCS1    ;GO WITH
9020                                                               ;62 IN RHCS1 FOR WRITE HEADER AND DATA
9021                                                              ;WITH INTERRUPT ENABLED
9022
9023                                      ;*ONE REVOLUTION = 16670 MICRO SEC., ONE SECTOR = 760
9024                                      ;*MICRO SEC. MAX TIME ALLOWED = ONE REVOLUTION + HEAD
9025                                      ;*SWITCH + 2 SECTORS, MIN TIME ALLOWED = SECTOR (FIRST CASE)
9026                                      ;*IF THERE IS A FAILURE HERE HALT PROGRAM AFTER ERROR WITH
9027                                      ;*SWITCH 15 AND SEE CURRENT CYLINDER REGISTER TO DETERMINE
9028                                      ;*WHAT CYLINDER IS FAILING
9029
9030
9031  031244  104413                      WAT                     ;WAIT FOR RDY BIT TO SET
9032  031246  002200                      RHCS1                   ;WAIT FOR RHCS1 REGISTER
9033  031250  000200                      RDY                     ;WAIT FOR RDY BIT IN RHCS1 REGISTER
9034  031252  003162                      1650.                   ;ALLOW 16500 MICRO SECONDS
9035  031254  001572                      890.                    ;RDY MUST SET BETWEEN
9036                                                              ;7600 AND 25400 MICRO SECONDS
9037
9038                                      ;*NOW SECTOR 0 OF ONE TRACK HAS BEEN WRITTEN CHECK COMPOSIT
9039                                      ;*ERROR BIT TO BE SURE NO ERRORS HAPPENED
9040
9041                                      ;*SAVE REGISTERS IN SAVE TABLE
9042  031256  004737  035236              JSR     PC,@#PUTREG
9043
9044  031262  032737  040000  002304      BIT     #ERR,@#DS1      ;ANY DISK ERRORS
9045  031270  001004                      BNE     9$              ;BRANCH IF YES
9046  031272  032737  040000  002262      BIT     #TRE,@#CS1      ;ANY RH ERRORS
9047  031300  001401                      BEQ     4$              ;BRANCH IF NO
9048
9049  031302  104066              9$:     ERROR   66              ;SOME ERRORS OCCURRED
9050                                                              ;WHILE DOING WRITE HEADER
9051                                                              ;AND DATA
9052
9053                                      ;*THE FOLLOWING 3 ADDS SETS UP FOR NEXT TRACK WRITING
9054
9055  031304  062737  000400  031146  4$:  ADD     #400,@#1$+12    ;NEXT TRACK FOR HEADER
9056  031312  062737  000040  031164      ADD     #40,@#2$+10     ;NEXT TRACK FOR DATA
```

```
 9057  031320  062737  000400  031174          ADD     #400,@#3$+6     ;NEXT TRACK FOR COMMAND
 9058
 9059  031326  005337  001200                  DEC     @#$TMP1         ;COUNT 19 TRACKS
 9060  031332  001300                          BNE     1$
 9061
 9062                                          ;*THE FOLLOWING CLEARS SETS UP FOR READ HEADER AND DATA
 9063  031334  005037  031414                  CLR     @#SST3+12       ;START WITH SECTOR/TRACK = 0
 9064  031340  005037  031432                  CLR     @#SST4+10       ;START WITH DATA = 0
 9065  031344  005037  031442                  CLR     @#SST5+6        ;START WITH 0 FOR COMMAND
 9066
 9067
 9068  031350  004737  033314                  JSR     PC,@#CLDISK     ;SET R1-RHCS1, R2-RHCS2
 9069                                                                  ;R3-RHDS1, R4-RHER1
 9070                                                                  ;GIVE RH-11 INITIALIZE
 9071                                                                  ;SETUP UNIT NUMBER
 9072
 9073                                          ;*SET UP FOR READ HEADER AND DATA
 9074  031354  012737  000023  001200  SST1:   MOV     #19.,@#$TMP1    ;19 TRACKS TO BE READ
 9075
 9076                                          ;*FILL READ INTO BUFFER WITH ALL ONES
 9077  031362                          SST2:
 9078
 9079  031362  004037  033164                  JSR     R0,@#CLAREA     ;CLEAR 260. WORDS, FROM REINTO
 9080  031366  003434                          REINTO                  ;STARTING FROM REINTO
 9081  031370  000404                          260.                    ;260. WORDS
 9082  031372  177777                          -1                      ;FILL WITH -1
 9083
 9084  031374  013737  031362  001110          MOV     @#SST2,@#$LPERR ;SET LOOP POINT
 9085
 9086                                          ;*FILL WRITE FROM BUFFER WITH EXPECTED HEADER
 9087  031402                          SST3:
 9088
 9089  031402  004037  033140                  JSR     R0,@#FLHEAD     ;SAVE HEADER DATA IN WRFROM
 9090  031406  002370                          WRFROM                  ;LOCATION WHERE SAVED
 9091  031410  000004                          4                       ;NUMBER OF WORDS SAVED
 9092  031412  010000                          10000                   ;FIRST DATA WORD
 9093  031414  000000                          0                       ;SECOND DATA WORD
 9094  031416  000000                          0                       ;THIRD DATA WORD
 9095  031420  000000                          0                       ;FOURTH DATA WORD
 9096
 9097                                          ;*FILL WRITE FROM BUFFER WITH EXPECTED DATA
 9098  031422                          SST4:
 9099
 9100  031422  004037  033164                  JSR     R0,@#CLAREA     ;CLEAR 256. WORDS, FROM WRFROM+<4*2>
 9101  031426  002400                          WRFROM+<4*2>                    ;STARTING FROM WRFROM+<4*2>
 9102  031430  000400                          256.                    ;256. WORDS
 9103  031432  000000                          <0.*2000>!<0*40>!0                      ;FILL WITH <0.*2000>!<0*40>!0
 9104
 9105
 9106                                          ;*FILL COMMAND FOR READ HEADER AND DATA
 9107  031434                          SST5:
 9108
 9109  031434  004037  035276                  JSR     R0,@#RUN        ;SETUP TO RUN FOR DATA COMMAND
 9110  031440  000000                          0                       ;CYLINDER 0
 9111  031442     000                          .BYTE   0               ;SECTOR 0
 9112  031443     000                          .BYTE   0               ;TRACK 0
```

```
 9113  031444  177374                          -256.-4                 ;WORD COUNT (DATA) = 256. +
 9114                                                                  ;4 HEADER WORDS
 9115  031446  003434                          REINTO                  ;BUS ADDRESS
 9116                                                                  ;STARTING ADDRESS OF DATA
 9117                                                                  ;BUFFER = REINTO
 9118  031450  000000                          0                       ;DO NOT INHIBIT BUS ADDRESS INCREMENT
 9119  031452  014000                          ECI!FMT22                       ;16 BITS PER WORD FORMAT
 9120                                                                  ;INHIBIT ECC CORRECTION
 9121                                                                  ;DO NOT INHIBIT HEADER COMPARE
 9122  031454  002350                          REFOR                   ;GET READY TO DO A REFOR
 9123                                                                  ;READ HEADER AND DATA WITH 72 IN RHCS1
 9124
 9125
 9126  031456  004737  033374                  JSR     PC,@#CHECKT     ;CHECKS DVA,RDY,MOL,DPR,DRY AND VV = 1
 9127                                                                  ;AND THAT ALL STATUS BITS ARE = 0
 9128  031462  104401  057465                  TYPE    ,CPHALT         ;CANNOT CONTINUE TESTING IF NOT
 9129  031466  000000                          HALT                    ;STOP TEST
 9130
 9131  031470  013777  004506  150470          MOV     @#RP4VEC,@RPVEC ;SET RP04 VECTOR ADDRESS
 9132                                                                  ;TO 'TIME1' IF P-CLOCK IS PRESENT
 9133                                                                  ;OR TO 'TIME2' IF P-CLOCK IS NOT THERE
 9134                                                                  ;'TIME' WILL ONLY SAVE
 9135                                                                  ;CURRENT CYLINDER ADDRESS
 9136                                                                  ;AND LOOK AHEAD REGISTERS
 9137
 9138
 9139  031476  013746  002350                  MOV     @#REFOR,-(SP)   ;GET READY TO MOVE COMMAND
 9140  031502  052716  000101                  BIS     #GO!IE,(SP)     ;GET READY TO SET GO AND
 9141                                                                  ;ENABLE INTERRUPT
 9142  031506  012677  150466                  MOV     (SP)+,@RHCS1    ;GO WITH
 9143                                                                    ;72 IN RHCS1 FOR READ DATA
 9144                                                                  ;WITH INTERRUPT ENABLED
 9145
 9146
 9147  031512  104413                          WAT                     ;WAIT FOR RDY BIT TO SET
 9148  031514  002200                          RHCS1                   ;WAIT FOR RHCS1 REGISTER
 9149  031516  000200                          RDY                     ;WAIT FOR RDY BIT IN RHCS1 REGISTER
 9150  031520  003162                          1650.                   ;ALLOW 16500 MICRO SECONDS
 9151  031522  001572                          890.                    ;RDY MUST SET BETWEEN
 9152                                                                  ;7600 AND 25400 MICRO SECONDS
 9153
 9154                                          ;*NOW SECTOR 0 OF ONE TRACK HAS BEEN READ CHECK COMPOSIT
 9155                                          ;*ERROR BIT TO BE SURE NO ERROR HAPPENED   .
 9156
 9157                                          ;*SAVE REGISTERS IN SAVE TABLE
 9158  031524  004737  035236                  JSR     PC,@#PUTREG
 9159
 9160  031530  032737  040000  002304          BIT     #ERR,@#DS1      ;ANY DISK ERRORS
 9161  031536  001004                          BNE     10$             ;BRANCH IF YES
 9162  031540  032737  040000  002262          BIT     #TRE,@#CS1      ;ANY RH ERRORS
 9163  031546  001401                          BEQ     11$             ;BRANCH IF NO
 9164
 9165  031550  104066                  10$:    ERROR   66              ;SOME ERRORS OCCURRED
 9166                                                                  ;WHILE DOING READ
 9167                                                                  ;HEADER AND DATA
 9168
```

```
 9169                                          ;*NOW THE READ DATA WILL BE COMPARED DATA IN EACH SECTOR
 9170                                          ;*IS UNIQUE IF PROGRAM IS HALTED ON ERROR THEN LOOK AT
 9171                                          ;*RHDST TO GET WHAT TRACK IS IN ERROR. LOOKING AT THE DATA
 9172                                          ;*BITS NO 4,5,6,7,8 IN GOOD DATA ALSO GIVES TRACK NUMBER
 9173                                          ;*IN GOOD DATA ALSO GIVES TRACK NUMBER
 9174
 9175  031552                          11$:
 9176
 9177  031552  004037  035342                 JSR     R0,@#COMPAR     ;COMPARE TWO BLOCKS OF MEMORY
 9178  031556  002370                         WRFROM                  ;GOOD DATA STARTS FROM WRFROM
 9179  031560  003434                         REINTO                  ;TEST DATA STARTS FROM REINTO
 9180  031562  000404                         260.                    ;260., WORDS TO BE COMPARED
 9181  031564  031570                         12$                     ;RETURN TO 12$ ON ERROR
 9182  031566  031574                         13$                     ;RETURN TO 13$ ON NO ERROR
 9183
 9184
 9185                                                                  ;BITS 4,5,6,7,8
 9186  031570  104067                  12$:   ERROR   67              ;READ HEADER AND DATA
 9187  031572  000207                         RTS     PC              ;ERROR
 9188                                                                  ;HEAD SELECTION ERROR
 9189                                                                  ;DATA READ GIVES NATURE
 9190                                                                  ;OF ERROR
 9191                                                                  ;EXCEPT FOR THE
 9192                                                                  ;FOUR  HEADER WORDS
 9193                                                                  ;THE BITS 4,5,6,7,8
 9194                                                                  ;GIVE THE TRACK NUMBER
 9195
 9196                                          ;*NOW INCREMENT TO READ NEXT TRACK
 9197
 9198  031574  062737  000400  031414  13$:    ADD     #400,@#SST3+12  ;NEXT TRACK FOR HEADER
 9199  031602  062737  000040  031432          ADD     #40,@#SST4+10   ;NEXT TRACK FOR DATA
 9200  031610  062737  000400  031442          ADD     #400,@#SST5+6   ;NEXT TRACK FOR COMMAND
 9201
 9202  031616  005337  001200                 DEC     @#$TMP1         ;COUNT 19 TRACKS
 9203  031622  001001                         BNE     5$
 9204
 9205  031624  000402                         BR      TST37           ;TO NEXT TEST
 9206
 9207  031626  000137  031362          5$:    JMP     @#SST2          ;JUMP BACK
```

```
                                ;;*****************************************************************
                                ;*TEST 37       DIFFERENCE LINES
                                ;*
                                ;*      A WRITE HEADER AND DATA WILL BE DONE ON ALL CYLINDERS
                                ;*      UP TO 256./512. ON SECTOR ZERO, TRACK ZERO. THE DATA WILL
                                ;*      BE THE CYLINDER NUMBER
                                ;*
                                ;*      THEN A RECALIBRATE AND READ HEADER AND DATA WILL BE DONE
                                ;*      ON CYLINDERS 0,1,2,4,8,16,32,64,128,256 & 512.
                                ;*
                                ;*      DATA WILL BE CHECKED.
                                ;*      ON AN ERROR, LOOPING WILL BE DONE ON THE
                                ;*      READ ONLY
                                ;*
                                ;;*****************************************************************
031632  000004                  TST37:  SCOPE
031634  012706  001000                  MOV     #STACK,SP       ;RESET STACK
031640  012737  000037  004504          MOV     #37,@#TSTNM     ;SAVE TEST NUMBER
031646  004737  033314                  JSR     PC,@#CLDISK     ;SET R1-RHCS1, R2-RHCS2
                                                                ;R3-RHDS1, R4-RHER1
                                                                ;GIVE RH-11 INITIALIZE
                                                                ;SETUP UNIT NUMBER
031652  004737  033374                  JSR     PC,@#CHECKT     ;CHECKS DVA,RDY,MOL,DPR,DRY AND VV = 1
                                                                ;AND THAT ALL STATUS BITS ARE = 0
031656  104401  057465                  TYPE    ,CPHALT         ;CANNOT CONTINUE TESTING IF NOT
031662  000000                          HALT                    ;STOP TEST

                                ;*SET UP TO INITIALIZE TEST FROM CYLINDER 0, TRACK 0,
                                ;*SECTOR 0
031664  012737  010000  031776  MOV     #10000,@#1$+10  ;CYLINDER HEADER DATA
031672  005037  032016          CLR     @#2$+10         ;DATA
031676  005037  032024          CLR     @#3$+4          ;CYLINDER COMMAND RHCA

                                ;*THIS IS TO GET THE HEADS TO CYLINDER ZERO
031702  013777  004506  150256  MOV     @#RP4VEC,@RPVEC ;SET RP04 VECTOR ADDRESS
                                                        ;TO 'TIME1' IF P-CLOCK IS PRESENT
                                                        ;OR TO 'TIME2' IF P-CLOCK IS NOT THERE
                                                        ;'TIME' WILL ONLY SAVE
                                                        ;CURRENT CYLINDER ADDRESS
                                                        ;AND LOOK AHEAD REGISTERS

031710  013746  002326          MOV     @#RECALI,-(SP)  ;GET READY TO MOVE COMMAND
031714  052716  000101          BIS     #GO!IE,(SP)     ;GET READY TO SET GO AND
                                                        ;ENABLE INTERRUPT
031720  012677  150254          MOV     (SP)+,@RHCS1    ;GO WITH
                                                         ;6 IN RHCS1 FOR RECALIBRATE
                                                        ;WITH INTERRUPT ENABLED


031724  104413                  WAT                     ;WAIT FOR DRY BIT TO SET
031726  002222                  RHDS1                   ;WAIT FOR RHDS1 REGISTER
031730  000200                  DRY                     ;WAIT FOR DRY BIT IN RHDS1 REGISTER
```

```
 9264  031732  060650                          25000.                  ;ALLOW 250000 MICRO SECONDS
 9265  031734  060650                          25000.                  ;DRY MUST SET BETWEEN
 9266                                                                  ;00 AND 500000 MICRO SECONDS
 9267
 9268                                          ;*THE DRIVE TYPE IS CHECKED AND THE APPROPRIATE MAX.
 9269                                          ;*CYLINDER DIFFERENCE IS SET UP
 9270
 9271                                  ;;**************************************************************
 9272  031736  005737  004636                 TST     @#RP06  ;TEST FOR RP06 DRIVE
 9273  031742  001404                         BEQ     14$             ;TREAT UNIT AS AN RP04
 9274                                                                  ;TREAT AS AN RP06
 9275                                  ;;**************************************************************
 9276
 9277  031744  012737  001001  001200         MOV     #513.,@#$TMP1   ;513 CYLINDERS
 9278  031752  000403                         BR      15$             ;CONTINUE
 9279
 9280  031754  012737  000401  001200 14$:    MOV     #257.,@#$TMP1   ;257 CYLINDERS
 9281  031762                         15$:                             ;CONTINUE WITH TEST
 9282
 9283
 9284  031762  004737  033314                 JSR     PC,@#CLDISK     ;SET R1-RHCS1, R2-RHCS2
 9285                                                                  ;R3-RHDS1, R4-RHER1
 9286                                                                  ;GIVE RH-11 INITIALIZE
 9287                                                                  ;SETUP UNIT NUMBER
 9288
 9289                                          ;*FILL WRITE FROM BUFFER WITH HEADER
 9290  031766                         1$:
 9291
 9292  031766  004037  033140                 JSR     R0,@#FLHEAD     ;SAVE HEADER DATA IN WRFROM
 9293  031772  002370                         WRFROM                  ;LOCATION WHERE SAVED
 9294  031774  000004                         4                       ;NUMBER OF WORDS SAVED
 9295  031776  010000                         10000                   ;FIRST DATA WORD
 9296  032000  000000                         0                       ;SECOND DATA WORD
 9297  032002  000000                         0                       ;THIRD DATA WORD
 9298  032004  000000                         0                       ;FOURTH DATA WORD
 9299
 9300                                          ;*FILL WRITE FROM BUFFER WITH DATA
 9301  032006                         2$:
 9302
 9303  032006  004037  033164                 JSR     R0,@#CLAREA     ;CLEAR 256. WORDS, FROM WRFROM+10
 9304  032012  002400                         WRFROM+10                       ;STARTING FROM WRFROM+10
 9305  032014  000400                         256.                    ;256. WORDS
 9306  032016  000000                         0                       ;FILL WITH 0
 9307
 9308
 9309                                          ;*THE WRITE HEADER AND DATA COMMAND WILL BE LOADED
 9310  032020                         3$:
 9311
 9312  032020  004037  035276                 JSR     R0,@#RUN        ;SETUP TO RUN FOR DATA COMMAND
 9313  032024  000000                         0                       ;CYLINDER 0
 9314  032026     000                         .BYTE   0               ;SECTOR 0
 9315  032027     000                         .BYTE   0               ;TRACK 0
 9316  032030  177374                         -256.-4                 ;WORD COUNT (DATA) = 256. +
 9317                                                                  ;4 HEADER WORDS
 9318  032032  002370                         WRFROM                  ;BUS ADDRESS
 9319                                                                  ;STARTING ADDRESS OF DATA
```

```
                                                        ;BUFFER = WRFROM
032034  000000                    0                     ;DO NOT INHIBIT BUS ADDRESS INCREMENT
032036  010000                    FMT22                 ;16 BITS PER WORD FORMAT
                                                        ;DO NOT INHIBIT ECC CORRECTION
                                                        ;DO NOT INHIBIT HEADER COMPARE
032040  002344                    WRIFOR                ;GET READY TO DO A WRIFOR
                                                ;WRITE HEADER AND DATA WITH 62 IN RHCS1

032042  004737  033374            JSR     PC,@#CHECKT   ;CHECKS DVA,RDY,MOL,DPR,DRY AND VV = 1
                                                        ;AND THAT ALL STATUS BITS ARE = 0
032046  104401  057465            TYPE    ,CPHALT       ;CANNOT CONTINUE TESTING IF NOT
032052  000000                    HALT                  ;STOP TEST

032054  013777  004506  150104    MOV     @#RP4VEC,@RPVEC ;SET RP04 VECTOR ADDRESS
                                                        ;TO 'TIME1' IF P-CLOCK IS PRESENT
                                                        ;OR TO 'TIME2' IF P-CLOCK IS NOT THERE
                                                        ;'TIME' WILL ONLY SAVE
                                                        ;CURRENT CYLINDER ADDRESS
                                                        ;AND LOOK AHEAD REGISTERS

032062  013746  002344            MOV     @#WRIFOR,-(SP) ;GET READY TO MOVE COMMAND
032066  052716  000101            BIS     #GO!IE,(SP)   ;GET READY TO SET GO AND
                                                        ;ENABLE INTERRUPT
032072  012677  150102            MOV     (SP)+,@RHCS1  ;GO WITH
                                                         ;62 IN RHCS1 FOR WRITE HEADER AND DATA
                                                        ;WITH INTERRUPT ENABLED

                                ;*ONE REVOLUTION = 16670 MICRO SECONDS, ONE SECTOR = 760
                                ;*MICRO SECONDS, ONE SEEK = 7000 MICRO SECONDS.
                                ;*MAX TIME = 1 REVOLUTION + 1 SEEK + 2 SECTORS
                                ;*MIN TIME = 1 SECTOR


032076  104413                    WAT                   ;WAIT FOR RDY BIT TO SET
032100  002200                    RHCS1                 ;WAIT FOR RHCS1 REGISTER
032102  000200                    RDY                   ;WAIT FOR RDY BIT IN RHCS1 REGISTER
032104  002354                    1260.                 ;ALLOW 12600 MICRO SECONDS
032106  002354                    1260.                 ;RDY MUST SET BETWEEN
                                                        ;00 AND 25200 MICRO SECONDS

                                ;*NOW ONE SECTOR WRITE IS COMPLETE.  CHANGES WILL BE MADE
                                ;*FOR THE NEXT SECTOR, THEN THE ABOVE WILL BE REPEATED
                                ;*UNTIL CYLINDER 256./512. IS REACHED
032110  005237  031776            INC     @#1$+10       ;CYLINDER HEADER DATA
032114  005237  032016            INC     @#2$+10       ;DATA
032120  005237  032024            INC     @#3$+4        ;CYLINDER COMMAND (RHCA)
032124  005337  001200            DEC     @#$TMP1       ;COUNT DOWN FOR 256./512. CYLINDERS
032130  001316                    BNE     1$            ;DO NEXT WRITE IF 256./512. NOT DONE

                                ;*NOW ALL 256./512. CYLINDERS HAVE CYLINDER NUMBER WRITTEN
                                ;*AS DATA ON SECTOR 0, TRACK 0. NOW A RECALIBRATE, FOLLOWED
                                ;*BY READ HEADER AND DATA, THEN A CHECK WILL BE DONE ON
                                ;*CYLINDER 0,1,2,4,8,16,32,64,128,256,512, AND 0
```

```
 9376  032132  013737  032162  001110          MOV     @#4$,@#$LPERR   ;LOOP ON ERROR
 9377  032140  005037  001200                  CLR     @#$TMP1         ;CYLINDER COUNTER
```

```
                                        ;*INITIALIZE, RECALIBRATE, AND READ CYLINDERS
                                    ;;*****************************************************************
                                        ;*SETUP FOR CYLINDER 0
032144  012737  010000  032244          MOV     #10000,@#5$+10  ;CYLINDER HEADER (DATA)
032152  005037  032264                  CLR     @#6$+10         ;DATA
032156  005037  032272                  CLR     @#7$+4          ;CYLINDER COMMAND (RHCA)
032162                              4$:
032162  004737  033314                  JSR     PC,@#CLDISK     ;SET R1-RHCS1, R2-RHCS2
                                                                ;R3-RHDS1, R4-RHER1
                                                                ;GIVE RH-11 INITIALIZE
                                                                ;SETUP UNIT NUMBER
032166  013777  004506  147772          MOV     @#RP4VEC,@RPVEC ;SET RP04 VECTOR ADDRESS
                                                                ;TO 'TIME1' IF P-CLOCK IS PRESENT
                                                                ;OR TO 'TIME2' IF P-CLOCK IS NOT THERE
                                                                ;'TIME' WILL ONLY SAVE
                                                                ;CURRENT CYLINDER ADDRESS
                                                                ;AND LOOK AHEAD REGISTERS
032174  013746  002326                  MOV     @#RECALI,-(SP)  ;GET READY TO MOVE COMMAND
032200  052716  000101                  BIS     #GO!IE,(SP)     ;GET READY TO SET GO AND
                                                                ;ENABLE INTERRUPT
032204  012677  147770                  MOV     (SP)+,@RHCS1    ;GO WITH
                                                                 ;6 IN RHCS1 FOR RECALIBRATE
                                                                ;WITH INTERRUPT ENABLED

032210  104413                          WAT                     ;WAIT FOR DRY BIT TO SET
032212  002222                          RHDS1                   ;WAIT FOR RHDS1 REGISTER
032214  000200                          DRY                     ;WAIT FOR DRY BIT IN RHDS1 REGISTER
032216  060650                          25000.                  ;ALLOW 250000 MICRO SECONDS
032220  060650                          25000.                  ;DRY MUST SET BETWEEN
                                                                ;00 AND 500000 MICRO SECONDS

                                        ;*CLEAR READ INTO BUFFER WITH ALL ONES

032222  004037  033164                  JSR     R0,@#CLAREA     ;CLEAR 260. WORDS, FROM REINTO
032226  003434                          REINTO                  ;STARTING FROM REINTO
032230  000404                          260.                    ;260. WORDS
032232  177777                          -1                      ;FILL WITH -1

                                        ;*FILL WRITE FROM BUFFER WITH EXPECTED HEADER
032234                              5$:
032234  004037  033140                  JSR     R0,@#FLHEAD     ;SAVE HEADER DATA IN WRFROM
032240  002370                          WRFROM                  ;LOCATION WHERE SAVED
032242  000004                          4                       ;NUMBER OF WORDS SAVED
032244  010000                          10000                   ;FIRST DATA WORD
032246  000000                          0                       ;SECOND DATA WORD
032250  000000                          0                       ;THIRD DATA WORD
032252  000000                          0                       ;FOURTH DATA WORD
```

```
9434  032254                          6$:
9435
9436  032254  004037  033164                  JSR     R0,@#CLAREA     ;CLEAR 256. WORDS, FROM WRFROM+10
9437  032260  002400                  WRFROM+10                       ;STARTING FROM WRFROM+10
9438  032262  000400                  256.                    ;256. WORDS
9439  032264  000000                  0                       ;FILL WITH 0
9440
9441
9442                                  ;*FILL READ HEADER AND DATA COMMAND
9443  032266                          7$:
9444
9445  032266  004037  035276                  JSR     R0,@#RUN        ;SETUP TO RUN FOR DATA COMMAND
9446  032272  000000                  0                       ;CYLINDER 0
9447  032274     000                  .BYTE   0               ;SECTOR 0
9448  032275     000                  .BYTE   0               ;TRACK 0
9449  032276  177374                  -256.-4                 ;WORD COUNT (DATA) = 256. +
9450                                                          ;4 HEADER WORDS
9451  032300  003434                  REINTO                  ;BUS ADDRESS
9452                                                          ;STARTING ADDRESS OF DATA
9453                                                          ;BUFFER = REINTO
9454  032302  000000                  0                       ;DO NOT INHIBIT BUS ADDRESS INCREMENT
9455  032304  014000                  ECI!FMT22                       ;16 BITS PER WORD FORMAT
9456                                                          ;INHIBIT ECC CORRECTION
9457                                                          ;DO NOT INHIBIT HEADER COMPARE
9458  032306  002350                  REFOR                   ;GET READY TO DO A REFOR
9459                                                          ;READ HEADER AND DATA WITH 72 IN RHCS1
9460
9461
9462  032310  004737  033374                  JSR     PC,@#CHECKT     ;CHECKS DVA,RDY,MOL,DPR,DRY AND VV = 1
9463                                                          ;AND THAT ALL STATUS BITS ARE = 0
9464  032314  104401  057465                  TYPE    ,CPHALT         ;CANNOT CONTINUE TESTING IF NOT
9465  032320  000000                  HALT                    ;STOP TEST
9466
9467  032322  013777  004506  147636          MOV     @#RP4VEC,@RPVEC ;SET RP04 VECTOR ADDRESS
9468                                                          ;TO 'TIME1' IF P-CLOCK IS PRESENT
9469                                                          ;OR TO 'TIME2' IF P-CLOCK IS NOT THERE
9470                                                          ;'TIME' WILL ONLY SAVE
9471                                                          ;CURRENT CYLINDER ADDRESS
9472                                                          ;AND LOOK AHEAD REGISTERS
9473
9474                                  ;*ONE SECTOR = 760 MICRO SECONDS, ONE REVOLUTION =
9475                                  ;*16670 MICRO SECONDS, MAX SEEK = 52000 MICRO SECONDS
9476                                  ;*MAX TIME = ONE REV + 1 SEEK + 1 SECTOR
9477                                  ;*MIN TIME = 1 SECTOR
9478
9479
9480  032330  013746  002350          MOV     @#REFOR,-(SP)   ;GET READY TO MOVE COMMAND
9481  032334  052716  000101          BIS     #GO!IE,(SP)     ;GET READY TO SET GO AND
9482                                                          ;ENABLE INTERRUPT
9483  032340  012677  147634          MOV     (SP)+,@RHCS1    ;GO WITH
9484                                                            ;72 IN RHCS1 FOR READ DATA
9485                                                          ;WITH INTERRUPT ENABLED
9486
9487
9488  032344  104413                  WAT                     ;WAIT FOR RDY BIT TO SET
9489  032346  002200                  RHCS1                   ;WAIT FOR RHCS1 REGISTER
```

```
9490  032350  000200                        RDY                     ;WAIT FOR RDY BIT IN RHCS1 REGISTER
9491  032352  006620                        3472.                   ;ALLOW 34720 MICRO SECONDS
9492  032354  006620                        3472.                   ;RDY MUST SET BETWEEN
9493                                                                  ;00 AND 69440 MICRO SECONDS
9494
9495                                        ;*CHECK READ WORDS AS ALL READ COMMANDS HAVE BEEN CHECKED
9496
9497                                        ;*(DATA ERRORS MAY IMPLY "IMPLIED SEEK" ERRORS)
9498
9499
9500
9501  032356  004037  035342                JSR     R0,@#COMPAR     ;COMPARE TWO BLOCKS OF MEMORY
9502  032362  002370                        WRFROM                  ;GOOD DATA STARTS FROM WRFROM
9503  032364  003434                        REINTO                  ;TEST DATA STARTS FROM REINTO
9504  032366  000404                        260.                    ;260., WORDS TO BE COMPARED
9505  032370  032374                        8$                      ;RETURN TO 8$ ON ERROR
9506  032372  032400                        9$                      ;RETURN TO 9$ ON NO ERROR
9507
9508
9509  032374  104070              8$:       ERROR   70              ;READ HEADER AND DATA ERROR
9510  032376  000207                        RTS     PC              ;DATA GIVES EXPECTED CYLINDER
9511
9512                                        ;*NOW ONE CYLINDER HAS BEEN CHECKED.  CHANGES WILL BE MADE
9513                                        ;*TO READ THE NEXT CYLINDER AND THE ABOVE SECTOR READ WILL BE
9514                                        ;*REPEATED
9515
9516  032400  005737  001200        9$:       TST     @#$TMP1         ;IS IT ZERO ?
9517  032404  001003                        BNE     10$             ;BRANCH IF NOT ZERO
9518  032406  005237  001200                INC     @#$TMP1         ;ADD ONE IF = 0
9519  032412  000416                        BR      11$             ;PUT ONE IN CYLINDER
9520
9521                                        ;;*****************************************************************
9522  032414  005737  004636        10$:      TST     @#RP06  ;TEST FOR RP06 DRIVE
9523  032420  001404                        BEQ     16$             ;TREAT UNIT AS AN RP04
9524                                                                  ;TREAT AS AN RP06
9525                                        ;;*****************************************************************
9526
9527  032422  022737  001000  001200        CMP     #512.,@#$TMP1   ;IS IT PASSED 512 CYLINDERS YET ?
9528  032430  000403                        BR      17$             ;CONTINUE
9529  032432  022737  000400  001200  16$:   CMP     #256.,@#$TMP1   ;IS IT PASSED 256 CYLINDERS YET ?
9530  032440                        17$:                            ;CONTINUE
9531
9532  032440  101421                        BLOS    12$             ;YES, SO GO TO ZERO
9533  032442  063737  001200  001200        ADD     @#$TMP1,@#$TMP1 ;DOUBLE THE CYLINDER
9534  032450  013737  001200  032264  11$:   MOV     @#$TMP1,@#6$+10 ;MAKE CYLINDER ADDRESS THE DATA
9535  032456  013746  001200                MOV     @#$TMP1,-(SP)   ;GET CYLINDER NUMBER
9536  032462  052716  010000                BIS     #FMT22,(SP)     ;INCLUDE FORMAT BIT
9537  032466  012637  032244                MOV     (SP)+,@#5$+10   ;HEADER DATA (CYLINDER)
9538  032472  013737  001200  032272        MOV     @#$TMP1,@#7$+4  ;CYLINDER COMMAND (RHCA)
9539  032500  000137  032162                JMP     @#4$            ;RETURN TO RECALIBRATE
9540
9541                                        ;;*****************************************************************
9542  032504  005737  004636        12$:      TST     @#RP06  ;TEST FOR RP06 DRIVE
9543  032510  001405                        BEQ     18$             ;TREAT UNIT AS AN RP04
9544                                                                  ;TREAT AS AN RP06
9545                                        ;;*****************************************************************
```

```
 9546
 9547  032512  022737  002000  001200          CMP     #1024.,@#$TMP1  ;512 DONE YET ?
 9548  032520  001421                          BEQ     13$             ;OUT ---------------------------->
 9549  032522  000404                          BR      19$             ;CONTINUE
 9550  032524  022737  001000  001200  18$:    CMP     #512.,@#$TMP1   ;256 DONE YET ?
 9551  032532  001414                          BEQ     13$             ;OUT ---------------------------->
 9552  032534                          19$:                            ;CONTINUE
 9553
 9554  032534  063737  001200  001200          ADD     @#$TMP1,@#$TMP1 ;DOUBLE THE CYLINDER
 9555  032542  012737  010000  032244          MOV     #10000,@#5$+10  ;CYLINDER HEADER DATA
 9556  032550  005037  032264                  CLR     @#6$+10         ;DATA
 9557  032554  005037  032272                  CLR     @#7$+4          ;CYLINDER COMMAND (RHCA)
 9558  032560  000137  032162                  JMP     @#4$            ;RETURN TO THE RECALIBRATE
 9559
 9560  032564                          13$:                            ;END OF TEST
```

```
 9561
 9562                                   ;;*******************************************************************
 9563                                   ;;*******************************************************************
 9564                                   ;*TEST 40       END OF DRIVE
 9565
 9566                                   ;*      THIS IS THE END OF TEST FOR ONE DRIVE
 9567                                   ;*      IF THERE ARE MORE DRIVES, THEN THE PROGRAM
 9568                                   ;*      JUMPS TO TEST 5 FOR NEXT DRIVE TEST
 9569
 9570                                   ;*      END OF PASS IS REACHED ONLY AFTER ALL DRIVES ARE COMPLETE
 9571
 9572                                   ;;*******************************************************************
 9573  032564  000004                  TST40:  SCOPE
 9574  032566  012737  000001  001212          MOV     #1,$TIMES       ;;DO 1 ITERATION
 9575  032574  012737  000000  177776          MOV     #0,PS           ;REINSTATE PS TO 0
 9576  032602  104401  032610                  TYPE    ,65$            ;;TYPE ASCIZ STRING
 9577  032606  000425                          BR      64$             ;;GET OVER THE ASCIZ
 9578                                  ;;65$:  .ASCIZ  <15><12>/TOTAL ERRORS ON THIS PASS ON  UNIT NO./
 9579  032662                          64$:
 9580  032662  013746  004616                  MOV     @#UNIT,-(SP)    ;GET READY TO TYPE UNIT NUMBER
 9581  032666  104405                          TYPDS
 9582  032670  104401  032676                  TYPE    ,67$            ;;TYPE ASCIZ STRING
 9583  032674  000402                          BR      66$             ;;GET OVER THE ASCIZ
 9584                                  ;;67$:  .ASCIZ  / =/
 9585  032702                          66$:
 9586  032702  013746  001112                  MOV     @#$ERTTL,-(SP)  ;GET READY TO TYPE NUMBER OF ERRORS
 9587  032706  104405                          TYPDS
 9588  032710  005037  001112                  CLR     @#$ERTTL        ;CLEAR TOTAL NUMBER OF ERRORS
 9589  032714  005037  001102                  CLR     @#$TSTNM        ;CLEAR TEST NUMBER
 9590  032720  005737  004626                  TST     @#SELECT        ;STARTING FROM 200 ?
 9591  032724  001413                          BEQ     3$              ;TEST NEXT DRIVE IF SO
 9592                                                                  ;CONTINUE TESTING THIS ONE IF NOT
 9593
 9594  032726  005237  001100                  INC     @#$PASS         ;INCREASE PASS COUNT
 9595  032732  104401  033123                  TYPE    ,$ENDMG         ;TYPE END PASS #
 9596  032736  013746  001100                  MOV     @#$PASS,-(SP)   ;GET PASS NO.
 9597  032742  104405                          TYPDS                   ;TYPE IT OUT
 9598  032744  104401  033120                  TYPE    ,$ENULL
 9599  032750  000137  010142                  JMP     @#TST4          ;JUMP TEST 4 ---------------------->
 9600
 9601  032754  012737  177777  041170  3$:     MOV     #-1,@#PRITEM    ;CLEAR PREVIOUS ITEM NUMBER
 9602  032762  005337  004620                  DEC     @#NOUNITS       ;NO. OF UNITS PRESENT DECREMENTED
 9603  032766  001413                          BEQ     $EOP            ;BRANCH IF ALL DRIVES COMPLETE
 9604  032770  013700  004616                  MOV     @#UNIT,R0       ;UNIT UNDER TEST
 9605  032774  012701  004576                  MOV     #UNITS,R1       ;TABLE
 9606  033000  022100                  1$:     CMP     (R1)+,R0        ;IS THIS UNIT JUST TESTED ?
 9607  033002  001401                          BEQ     2$              ;CONTINUE IF YES
 9608  033004  000775                          BR      1$              ;INCREMENT IF NO
 9609  033006  011137  004616          2$:     MOV     (R1),@#UNIT     ;THIS IS NEXT UNIT
 9610  033012  000137  010142                  JMP     @#TST4          ;TEST THE NEXT DRIVE ---------------->
 9611
```

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



                                .SBTTL  END OF PASS ROUTINE

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

033016                          $EOP:
033016  000004                          SCOPE
033020  005037  001102                  CLR     $TSTNM          ;;ZERO THE TEST NUMBER
033024  005037  001212                  CLR     $TIMES          ;;ZERO THE NUMBER OF ITERATIONS
033030  005237  001100                  INC     $PASS           ;;INCREMENT THE PASS NUMBER
033034  042737  100000  001100          BIC     #100000,$PASS   ;;DON'T ALLOW A NEG. NUMBER
033042  005327                          DEC     (PC)+           ;;LOOP?
033044  000001                  $EOPCT: .WORD   1
033046  003022                          BGT     $DOAGN          ;;YES
033050  012737                          MOV     (PC)+,@(PC)+    ;;RESTORE COUNTER
033052  000001                  $ENDCT: .WORD   1
033054  033044                          $EOPCT
033056  104401  033123                  TYPE    ,$ENDMG         ;;TYPE "END PASS #"
033062  013746  001100                  MOV     $PASS,-(SP)     ;;SAVE $PASS FOR TYPEOUT
033066  104405                          TYPDS                   ;;GO TYPE--DECIMAL ASCII WITH SIGN
033070  104401  033120                  TYPE    ,$ENULL         ;;TYPE A NULL CHARACTER
033074  013700  000042          $GET42: MOV     @#42,R0         ;;GET MONITOR ADDRESS
033100  001405                          BEQ     $DOAGN          ;;BRANCH IF NO MONITOR
033102  000005                          RESET                   ;;CLEAR THE WORLD
033104  004710                  $ENDAD: JSR     PC,(R0)         ;;GO TO MONITOR
033106  000240                          NOP                     ;;SAVE ROOM
033110  000240                          NOP                     ;;FOR
033112  000240                          NOP                     ;;ACT11
033114                          $DOAGN:
033114  000137                          JMP     @(PC)+          ;;RETURN
033116  006222                  $RTNAD: .WORD   TST1
033120     377        377  000 $ENULL: .BYTE   -1,-1,0         ;;NULL CHARACTER STRING
033123     015  042412  042116 $ENDMG: .ASCIZ  <15><12>/END PASS #/
033130  050040  051501  020123
033136  000043
```

```
                                        ;THIS FILLS MEMORY WITH GIVEN DATA
                                        ;USED CHIEFLY FOR HEADER INFORMATION
                                        ;CALL IS
                                        ;       JSR     R0,@#FLHEAD     ;FILL HEADER
                                        ;       LOC                     ;LOCATION WHERE SAVED
                                        ;       XN                      ;NUMBER OF WORDS
                                        ;       XD1                     ;DATA REPEATED XN TIMES
                                        ;       XD2                     ;DATA REPEATED XN TIMES
                                        ;
                                        ;
                                        ;
033140                                  FLHEAD:
033140  010146                                  MOV     R1,-(SP)        ;;PUSH R1 ON STACK
033142  010246                                  MOV     R2,-(SP)        ;;PUSH R2 ON STACK
033144  012001                                  MOV     (R0)+,R1        ;R1 HAS ADDRESS OF WHERE TO SAVE
033146  012002                                  MOV     (R0)+,R2        ;R2 HAS NUMBER OF WORDS

                                                ;*NOW FILL DATA

033150  012021                          1$:     MOV     (R0)+,(R1)+     ;SAVE DATA
033152  005302                                  DEC     R2              ;DECREMENT COUNT
033154  001375                                  BNE     1$              ;BRANCH IF INCOMPLETE
033156  012602                                  MOV     (SP)+,R2        ;;POP STACK INTO R2
033160  012601                                  MOV     (SP)+,R1        ;;POP STACK INTO R1
033162  000200                                  RTS     R0




                                        ;THIS CLEARS ANY BLOCK OF MEMORY.
                                        ;FILLING IT WITH ANY DATA
                                        ;CALL IS
                                        ;       JSR     R0,@#CLAREA
                                        ;       F                       ;FROM
                                        ;       N                       ;NUMBER OF WORDS
                                        ;       D                       ;DATA  TO BE FILLED
                                        ;
                                        ;R1 WILL HAVE STARTING ADDRESS OF BLOCK TO BE FILLED
                                        ;R2 WILL HAVE NUMBER OF WORDS
                                        ;R3 WILL HAVE DATA

033164                                  CLAREA:
033164  010146                                  MOV     R1,-(SP)        ;;PUSH R1 ON STACK
033166  010246                                  MOV     R2,-(SP)        ;;PUSH R2 ON STACK
033170  010346                                  MOV     R3,-(SP)        ;;PUSH R3 ON STACK
033172  012001                                  MOV     (R0)+,R1        ;FROM
033174  012002                                  MOV     (R0)+,R2        ;NUMBER
033176  012003                                  MOV     (R0)+,R3        ;DATA
033200  010321                          1$:     MOV     R3,(R1)+        ;MOVE DATA
033202  005302                                  DEC     R2              ;COUNT
033204  001375                                  BNE     1$              ;BRANCH IF NOT COMPLETE
033206  012603                                  MOV     (SP)+,R3        ;;POP STACK INTO R3
033210  012602                                  MOV     (SP)+,R2        ;;POP STACK INTO R2
033212  012601                                  MOV     (SP)+,R1        ;;POP STACK INTO R1
```

```
 9715  033214  000200                        RTS     R0              ;RETURN TO MAIN PROGRAM
 9716
 9717
 9718
 9719
 9720
 9721
 9722                                ;THIS IS A SUBROUTINE TO FILL SAVED REGISTER LOCATION
 9723                                ;WITH GIVEN VALUE
 9724                                ;CALL IS
 9725                                ;       JSR     R0,@#FILLRE
 9726                                ;       RHXX            ;REGISTER NAME
 9727                                ;       D               ;DATA
 9728                                ;
 9729
 9730  033216                        FILLRE:
 9731  033216  010146                       MOV     R1,-(SP)        ;;PUSH R1 ON STACK
 9732  033220  010246                       MOV     R2,-(SP)        ;;PUSH R2 ON STACK
 9733  033222  012001                       MOV     (R0)+,R1        ;ADDRESS OF ADDRESS OF REGISTER
 9734  033224  012002                       MOV     (R0)+,R2        ;DATA
 9735  033226  162701  002172               SUB     #RHWC,R1        ;OFSET
 9736  033232  010261  004512               MOV     R2,SAVERE(R1)   ;DATA IS MOVED IN
 9737  033236  012602                       MOV     (SP)+,R2        ;;POP STACK INTO R2
 9738  033240  012601                       MOV     (SP)+,R1        ;;POP STACK INTO R1
 9739  033242  000200                       RTS     R0              ;RETURN TO MAIN PROGRAM
 9740
 9741
 9742
```

```
                                        ;THIS SUBROUTINE SETS UP FOR SEARCH
                                        ;CALL IS
                                        ;       JSR     R0,@#SRCH
                                        ;       C               ;CYLINDER
                                        ;.BYTE  S               ;SECTOR
                                        ;.BYTE  T               ;TRACK

033244  012077  146742          SRCH:   MOV     (R0)+,@RHCA     ;SET DESIRED CYLINDER ADDRESS
033250  012077  146730          SRCH:   MOV     (R0)+,@RHDST    ;SET DESIRED SECTOR/TRACK ADDRESS
033254  013777  002334  146716          MOV     @#SERCH,@RHCS1  ;GET READY FOR SEARCH
                                                                ;WITH 30 IN RHCS1
033262  000200                          RTS     R0


                                        ;THIS SUBROUTINE SETS UP FOR SEEK COMMANDS
                                        ;CALL IS
                                        ;       JSR     R0,@#SEEKCY
                                        ;       C               ;CYLINDER
                                        ;

033264  012077  146722          SEEKCY: MOV     (R0)+,@RHCA     ;SET DESIRED CYLINDER ADDRESS
033270  013777  002352  146702          MOV     @#SEECOM,@RHCS1 ;MOV 4 INTO RHCS1
033276  000200                          RTS     R0              ;RETURN TO MAIN PROGRAM
```

```
 9771
 9772                                   ;THIS SUBROUTINE SETS UP FOR OFFSET COMMANDS
 9773                                   ;CALL IS
 9774                                   ;       JSR     R0,@#OFSET
 9775                                   ;       0                               ;MICRO INCHES OFFSET
 9776
 9777  033300  052077  146704          OFSET:  BIS     (R0)+,@RHOF             ;SET OFFSET REGISTER
 9778  033304  013777  002354  146666          MOV     @#OFSETC,@RHCS1         ;MOV14 INTO RHCS1
 9779  033312  000200                          RTS     R0                      ;RETURN TO MAIN PROGRAM
 9780
 9781
 9782  033314  013701  002200          CLDISK: MOV     @#RHCS1,R1              ;R1 WILL BE CONTROL AND STATUS1
 9783  033320  013702  002176                  MOV     @#RHCS2,R2              ;R2 WILL BE CONTROL AND STATUS2
 9784  033324  013703  002222                  MOV     @#RHDS1,R3              ;R3 WILL BE DISK STATUS REGISTER1
 9785  033330  013704  002202                  MOV     @#RHER1,R4              ;R4 WILL BE ERROR REGISTER #1
 9786
 9787  033334  012712  000040                  MOV     #CLR,@R2                ;CLEAR ALL REG.
 9788  033340  013712  004616                  MOV     @#UNIT,@R2              ;REINSTATE UNIT NO.
 9789  033344  005011                          CLR     @R1                     ;CLEAR FUNCTION BITS
 9790  033346  000207                          RTS     PC
```

```
 9791
 9792
 9793
 9794
 9795                                   ;THIS CHECKS DEVICE AVAILABLE (DVA) AND READY (RDY) IN RHCS1
 9796                                   ;AND CHECKS MEDIUM ON LINE (MOL), DEVICE PRESENT (DPR), DEVICE READY (DRY) IN RHDS1
 9797
 9798                                   ;IT MAY CHECK VOLUME VALID (VV) IN RHDS1, DEPENDING ON ENTRY POINT
 9799
 9800
 9801  033350  000000                   PCJSR:  0                       ;PC OF JSR
 9802
 9803  033352  011637  033350           CHECK:  MOV     (SP),@#PCJSR    ;SAVE PC OF JSR+4
 9804  033356  162737  000004  033350           SUB     #4,@#PCJSR      ;GET PC OF JSR
 9805  033364  011346                           MOV     @R3,-(SP)       ;GET RHDS1
 9806  033366  052716  000100                   BIS     #VV,(SP)        ;DONT CHECK VV BIT
 9807  033372  000406                           BR      CHECKC          ;GOTO COMMON CHECK ROUTINE
 9808
 9809  033374  011637  033350           CHECKT: MOV     (SP),@#PCJSR    ;SAVE PC OF JSR+4
 9810  033400  162737  000004  033350           SUB     #4,@#PCJSR      ;GET PC OF JSR
 9811  033406  011346                           MOV     @R3,-(SP)       ;GET RHDS1 & DO VV CHECK AT 3$
 9812
 9813  033410  011146                   CHECKC: MOV     @R1,-(SP)       ;GET CS1
 9814  033412  042716  173577                   BIC     #173577,(SP)    ;CLEAR UNWANTED BITS
 9815  033416  022726  004200                   CMP     #DVA!RDY,(SP)+  ;RHCS1 SHOULD HAVE DEVICE AVAILABLE
 9816                                                                   ;AND BE READY
 9817  033422  001403                           BEQ     3$              ;BRANCH IF GOOD
 9818  033424  011137  001122                   MOV     @R1,@#$BDADR    ;BAD DATA REGISTER (RHCS1)
 9819  033430  104062                           ERROR   62              ;RHCS1 DID NOT HAVE DEVICE
 9820                                                                   ;AVAILABLE RIGHT AT THE START
 9821                                                                   ;ALL OTHER BITS SHOULD BE 0
 9822  033432  042716  102000           3$:     BIC     #ATA!LBT,(SP)   ;CLEAR UNWANTED BITS
 9823  033436  022726  010700                   CMP     #MOL!DPR!DRY!VV,(SP)+ ;RHDS1 SHOULD HAVE THESE SET
 9824  033442  001404                           BEQ     7$              ;BRANCH IF GOOD
 9825  033444  011337  001122                   MOV     @R3,@#$BDADR    ;BAD DATA IN REGISTER (RHDS1)
 9826  033450  104061                           ERROR   61              ;RHDS1 HAS SOME BITS OTHER
 9827                                                                   ;THAN MOL, DRY, DPR,VV SET
 9828                                                                   ;ALL OTHER BITS SHOULD BE 0
 9829  033452  000207                           RTS     PC              ;RETURN TO TEST AND HALT
 9830
 9831  033454  062716  000006           7$:     ADD     #6,(SP)         ;ADJUST STACK TO JUMP OVER HALT IN TEST
 9832  033460  000207                           RTS     PC              ;RETURN TO TEST AND CONTINUE
```

```
 9833
 9834                                        ;*THIS IS A SUBROUTINE TO SAVE REGISTERS
 9835                                        ;*IN THE REGISTER TABLE TO ANY LOCATION
 9836                                        ;*THE CALL IS
 9837                                        ;*JSR   R0,@#SAVER
 9838                                        ;*      F       ;FROM
 9839                                        ;*      T       ;TO
 9840                                        ;*      N       ;NUMBER OF WORDS SAVED
 9841                                        ;*F MUST ALWAYS BE RHCS1
 9842                                        ;*T MUST ALWAYS BE SAVRE
 9843
 9844  033462                          SAVER:
 9845  033462  010146                       MOV     R1,-(SP)        ;;PUSH R1 ON STACK
 9846  033464  010246                       MOV     R2,-(SP)        ;;PUSH R2 ON STACK
 9847  033466  010346                       MOV     R3,-(SP)        ;;PUSH R3 ON STACK
 9848  033470  012001                       MOV     (R0)+,R1        ;FROM
 9849  033472  012002                       MOV     (R0)+,R2        ;TO
 9850  033474  012003                       MOV     (R0)+,R3        ;NUMBER
 9851  033476  013122                 1$:   MOV     @(R1)+,(R2)+    ;SAVE REGISTER CONTENTS
 9852  033500  005303                       DEC     R3              ;COUNT
 9853  033502  001375                       BNE     1$              ;BRANCH IF NOT DONE
 9854  033504  012603                       MOV     (SP)+,R3        ;;POP STACK INTO R3
 9855  033506  012602                       MOV     (SP)+,R2        ;;POP STACK INTO R2
 9856  033510  012601                       MOV     (SP)+,R1        ;;POP STACK INTO R1
 9857  033512  000200                       RTS     R0
 9858
 9859
 9860
 9861
 9862
 9863
 9864
 9865
 9866
 9867
 9868
 9869
 9870
 9871
 9872
 9873                                        ;WHEN AN EVENT IS TO BE TIMED THE RP04 VECTORS TO "TIME 1"
 9874                                        ;PRIORITY OF PROCESS OR IS 4
 9875                                        ;PRIORITY OF TRAPS MUST BE 6
 9876                                        ;PRIORITY OF RP04 INTERRUPTS IS 7
 9877                                        ;
 9878
 9879  033514  005077  146524          TIME1:  CLR     @PCLCSR         ;STOP THE CLOCK
 9880  033520  017737  146524  033552          MOV     @PCLCTR,@#WAITTM ;GET TIME ON CLOCK
 9881  033526  017737  146502  004564  TIME2:  MOV     @RHCC,@#FINACC  ;GET CURRENT CYLINDER
 9882  033534  017737  146476  004562          MOV     @RHLA,@#FINALA  ;GET LOOK AHEAD
 9883  033542  000002                          RTI                     ;RETURN TO WAIT P OR WAIT.T
 9884
 9885
 9886
 9887                                        ;THIS IS A WAIT LOOP WHEN AN EVENT IS TO BE TIMED
 9888                                        ;THE CALL IS
```

```
                                        ;       WAT
                                        ;       A       ;ABSOLUTE REGISTER ADDRESS
                                        ;       B       ;BIT WAITED FOR
                                        ;       TA      ;TIME ALLOWED GIVEN IN 10 MICROSEC
                                        ;       TO      ;TOLERANCE PLUS/MINUS IN 10 MICROSEC
                                        ;
                                        ;R1-WILL HAVE TIME ALLOWED IN 10 MICRO SECONDS
                                        ;R2-WILL HAVE TOLERANCE PLUS/MINUS IN 10 MICRO SECONDS
                                        ;MINIMUM TIME THAT CAN BE MEASURED IS ABOUT 12 MICRO SECONDS
                                        ;FOR THE SLOWEST PROCESSOR

033544  000000                  WAITPC: 0                       ;WAT PC
033546  000000                  WAITRE: 0                       ;WAIT ON REGISTER ADDRESS
033550  000000                  WAITBT: 0                       ;WAIT ON BIT
033552  000000                  WAITTM: 0                       ;WAITED TIME
033554  005037  033552          WAIT.P: CLR     @#WAITTM        ;CLEAR WAITED TIME
033560  005077  146462                  CLR     @PCLBUF ;CLEAR COUNT SET BUFFER
033564  012777  000021  146452          MOV     #GO!BIT4,@PCLCSR ;COUNT UP, 100 KHZ, START CLOCK
033572  010046                          MOV     R0,-(SP)        ;;PUSH R0 ON STACK
033574  010146                          MOV     R1,-(SP)        ;;PUSH R1 ON STACK
033576  010246                          MOV     R2,-(SP)        ;;PUSH R2 ON STACK
033600  010346                          MOV     R3,-(SP)        ;;PUSH R3 ON STACK
033602  016600  000010                  MOV     10(SP),R0       ;R0 HAS ADDRESS OF NEXT LOCATION
033606  010037  033544                  MOV     R0,@#WAITPC     ;NOW WAITPC HAS WAT PC + 2
033612  162737  000002  033544          SUB     #2,@#WAITPC     ;WAT PC IS IN WAITPC
033620  013037  033546                  MOV     @(R0)+,@#WAITRE ;WAIT ON REGISTER ADDRESS
033624  012037  033550                  MOV     (R0)+,@#WAITBT  ;WAIT ON BIT
033630  012001                          MOV     (R0)+,R1        ;R1 HAS TIME IN 10 MSEC
033632  012002                          MOV     (R0)+,R2        ;R2 HAS TOLERANCE IN 10 MSEC
033634  010066  000010                  MOV     R0,10(SP)       ;RESTORE RETURN ON STACK

                                        ;*THIS SECTION WAITS FOR BIT, THROUGH TWO COUNT DOWNS

033640  013703  034012                  MOV     @#TIMCNT,R3     ;R3 IS A TEMPORARY COUNTER
033644  033777  033550  177674  1$:     BIT     @#WAITBT,@WAITRE ;IS REQUIRED BIT THERE
033652  001025                          BNE     4$              ;BRANCH IF YES
033654  005303                          DEC     R3              ;COUNT IF REQUIRED BIT NOT THERE
033656  001372                          BNE     1$
033660  013703  034012                  MOV     @#TIMCNT,R3     ;TEMPORARY COUNTER
033664  033777  033550  177654  2$:     BIT     @#WAITBT,@WAITRE ;IS REQUIRED BIT THERE
033672  001015                          BNE     4$              ;BRANCH IF YES
033674  005303                          DEC     R3              ;COUNT IF REQUIRED BIT NOT THERE
033676  001372                          BNE     2$
033700  017737  177642  001126          MOV     @WAITBT,@#$BDDAT ;REGISTER CONTENTS FOR TYPEOUT
033706  032777  000100  146264          BIT     #IE,@RPCS1      ;DID ANY INTERRUPT OCCUR
033714  001402                          BEQ     3$              ;BRANCH IF YES
033716  104001                          ERROR   1               ;RP04 DID NOT INTERRUPT
033720  000427                          BR      7$              ;OUT
033722  104002                  3$:     ERROR   2               ;RP04 INTERRUPTED BUT WAITED
                                                                ;ON BIT DID NOT OCCUR
                                                                ;EVEN AFTER TWO COUNT DOWNS
                                                                ;FROM 177777 TO 0
033724  000425                          BR      7$              ;OUT

                                        ;*NOW TIME AND TOLERANCE WILL BE CHECKED
033726  017737  177614  001126  4$:     MOV     @WAITRE,@#$BDDAT ;REGISTER CONTENTS FOR TYPEOUT
```

```
 9945  033734  032777  000100  146236          BIT     #IE,@RHCS1      ;DID ANY INTERRUPT OCCUR
 9946  033742  001402                          BEQ     5$              ;BRANCH IF YES
 9947  033744  104003                          ERROR   3               ;INTERRUPT DID NOT OCCUR EVEN
 9948                                                                  ;AFTER ONE BNE AND ONE MOV
 9949                                                                  ;OF THE WAITED ON BIT SETTING
 9950  033746  000414                          BR      7$              ;OUT
 9951  033750  160201                  5$:     SUB     R2,R1           ;R1 NOW HAS LOWER LIMIT OF TIME
 9952  033752  023701  033552                  CMP     @#WAITTM,R1     ;FOR GOOD RESULTS, WAITTM
 9953                                                                  ;MUST BE GREATER OR EQUAL
 9954                                                                  ;TORI
 9955  033756  103002                          BHIS    6$              ;BRANCH IF GOOD
 9956  033760  104004                          ERROR   4               ;BIT DID OCCUR BUT TIME
 9957                                                                  ;TAKEN IS BELOW LOWER LIMIT
 9958  033762  000406                          BR      7$              ;OUT
 9959
 9960  033764  060202                  6$:     ADD     R2,R2           ;DOUBLE TOLERANCE
 9961  033766  060201                          ADD     R2,R1           ;R1 NOW HAS UPPER LIMIT OF TIME
 9962  033770  020137  033552                  CMP     R1,@#WAITTM     ;FOR GOOD RESULTS, WAITTM
 9963                                                                  ;MUST BE LESS OR EQUAL TO R1
 9964  033774  103001                          BHIS    7$              ;BRANCH IF GOOD
 9965  033776  104004                          ERROR   4               ;BIT DID OCCUR BUT TIME TAKEN
 9966                                                                  ;IS ABOVE UPPER LIMIT
 9967  034000                          7$:
 9968  034000  012603                          MOV     (SP)+,R3        ;;POP STACK INTO R3
 9969  034002  012602                          MOV     (SP)+,R2        ;;POP STACK INTO R2
 9970  034004  012601                          MOV     (SP)+,R1        ;;POP STACK INTO R1
 9971  034006  012600                          MOV     (SP)+,R0        ;;POP STACK INTO R0
 9972  034010  000002                          RTI                     ;RETURN TO MAIN TEST
 9973
 9974
 9975
 9976
 9977
 9978
 9979                                  ;THIS IS A WAIT LOOP WHEN NO P-CLOCK IS AVAILABLE
 9980                                  ;NO TIMING IS DONE
 9981                                  ;CALL IS
 9982                                  ;       WAT
 9983                                  ;       A       ;ABSOLUTE REGISTER ADDRESS
 9984                                  ;       B       ;BIT WAITED FOR
 9985                                  ;       TA      ;TIME-NOT USED HERE
 9986                                  ;       TO      ;TIME-NOT USED HERE
 9987                                  ;R3-IS A TEMPORARY COUNTER
 9988
 9989  034012  177777                  TIMCNT: 177777                  ;COUNT FOR WAIT LOOP
 9990  034014  000025                  RPTCTR: 25                      ;COUNT FOR INTERRUPT WAIT (11/70 CPU)
 9991
 9992
 9993  034016                          WAIT.T:
 9994  034016  010046                          MOV     R0,-(SP)        ;;PUSH R0 ON STACK
 9995  034020  010346                          MOV     R3,-(SP)        ;;PUSH R3 ON STACK
 9996
 9997  034022  016600  000004                  MOV     4(SP),R0        ;R0 HAS ADDRESS OF NEXT LOCATION
 9998  034026  010037  033544                  MOV     R0,@#WAITPC     ;WAT PC +2 IS IN WAITPC
 9999  034032  162737  000002  033544          SUB     #2,@#WAITPC     ;WAT PC IS IN WAITPC
10000  034040  013037  033546                  MOV     @(R0)+,@#WAITRE ;WAIT ON REGISTER ADDRESS
```

```
10001  034044  012037  033550                  MOV     (R0)+,@#WAITBT  ;WAIT ON BIT
10002  034050  022020                          CMP     (R0)+,(R0)+     ;DUMP NEXT TWO WORDS-TA, TO
10003  034052  010066  000004                  MOV     R0,4(SP)        ;RESTORE RETURN ON STACK
10004
10005                                          ;*THIS HAS THE TWO COUNT DOWNS FROM 177777
10006
10007  034056  013703  034012                  MOV     @#TIMCNT,R3     ;R3 HAS TEMPORARY COUNT
10008  034062  033777  033550  177456  1$:     BIT     @#WAITBT,@WAITRE ;IS REQUIRED BIT THERE ?
10009  034070  001025                          BNE     4$              ;CHECK FOR THE INTERRUPT
10010  034072  005303                          DEC     R3              ;COUNT IF REQUIRED BIT NOT THERE
10011  034074  001372                          BNE     1$
10012  034076  013703  034012                  MOV     @#11MCNT,R3     ;SECOND COUNT DOWN FROM 177777
10013  034102  033777  033550  177436  2$:     BIT     @#WAITBT,@WAITRE ;IS REQUIRED BIT THERE ?
10014  034110  001015                          BNE     4$              ;CHECK FOR INTERRUPT
10015  034112  005303                          DEC     R3              ;COUNT IF REQUIRED BIT NOT THERE
10016  034114  001372                          BNE     2$
10017  034116  017737  177424  001126          MOV     @WAITRE,@#$BDDAT ;REGISTER CONTENTS FOR TYPEOUT
10018  034124  032777  000100  146046          BIT     #IE,@RHCS1      ;DID ANY INTERRUPT OCCUR ?
10019  034132  001402                          BEQ     3$              ;BRANCH IF YES
10020
10021  034134  104001                          ERROR   1               ;RP04 DID NOT INTERRUPT
10022                                                                  ;BIT DID NOT OCCUR
10023  034136  000417                          BR      5$              ;OUT ------------------------------->
10024
10025  034140  104002                  3$:     ERROR   2               ;RP04 INTERRUPTED BUT
10026                                                                  ;WAITED ON BIT DID NOT OCCUR
10027                                                                  ;EVEN AFTER TWO COUNT DOWNS
10028                                                                  ;FROM 177777 TO 0
10029  034142  000415                          BR      5$              ;OUT ------------------------------->
10030
10031                                          ;*BIT DID SET SO CHECK IF INTERRUPT OCCURRED
10032
10033                                          ;*THE AMOUNT OF TIME ALLOWED CAN BE CHANGED BY ALTERING LOCATION
10034                                          ;*''RPTCTR'' ABOVE
10035
10036  034144  013703  034014          4$:     MOV     @#RPTCTR,R3     ;LOAD COUNTER WITH COUNT
10037  034150  005303                  6$:     DEC     R3              ;COUNT DOWN ONE
10038  034152  001376                          BNE     6$              ;DO AGAIN IF NOT ZERO YET
10039
10040
10041  034154  032777  000100  146016          BIT     #IE,@RHCS1      ;DID ANY INTERRUPT OCCUR ?
10042  034162  001405                          BEQ     5$              ;BRANCH IF YES
10043  034164  017737  177356  001126          MOV     @WAITRE,@#$BDDAT ;REGISTER CONTENTS FOR TYPEOUT
10044  034172  104003                          ERROR   3               ;INTERRUPT DID NOT OCCUR
10045                                                                  ;EVEN AFTER ONE BNE OF
10046                                                                  ;THE WAITED ON BIT OCCURING
10047  034174  000400                          BR      5$              ;OUT ------------------------------->
10048
10049  034176                          5$:
10050  034176  012603                          MOV     (SP)+,R3        ;;POP STACK INTO R3
10051  034200  012600                          MOV     (SP)+,R0        ;;POP STACK INTO R0
10052  034202  000002                          RTI                     ;RETURN TO MAIN TEST
10053
```

```
10054
10055                                   ;THIS CHANGES REGISTER SAVED VALUE
10056                                   ;CALL IS
10057                                   ;       JSR     R0,@#CHREG
10058                                   ;       R                       ;REGISTER TO BE CHANGED
10059                                   ;       N                       ;NUMBER OF BITS TO BE CHANGED
10060                                   ;       NEW                     ;NEW VALUE OF BIT MUST BE 0 OR1
10061                                   ;       P                       ;POSITION OF BIT TO BE CHANGED
10062                                   ;NEW AND P WILL BE REPEATED N NUMBER OF TIMES
10063
10064   034204                          CHREG:
10065   034204  010146                          MOV     R1,-(SP)        ;;PUSH R1 ON STACK
10066   034206  010246                          MOV     R2,-(SP)        ;;PUSH R2 ON STACK
10067   034210  012001                          MOV     (R0)+,R1        ;R1 HAS ADDRESS OF ADDRESS OF REGISTER
10068   034212  012002                          MOV     (R0)+,R2        ;R2 HAS NUMBER OF CHANGES
10069   034214  162701  002172                  SUB     #RHWC,R1        ;R1 HAS OFSET OF REQUIRED REGISTER
10070   034220  005720                  1$:     TST     (R0)+           ;IS A BIC OR A BIS TO BE DONE
10071   034222  001403                          BEQ     2$              ;BRANCH IF A BIC IS REQUIRED
10072   034224  052061  004512                  BIS     (R0)+,SAVERE(R1) ;SET REQUIRED BIT
10073   034230  000402                          BR      3$              ;BRANCH TO DECREMENT COUNT
10074   034232  042061  004512          2$:     BIC     (R0)+,SAVERE(R1) ;CLEAR REQUIRED BIT
10075   034236  005302                  3$:     DEC     R2              ;DECREMENT NUMBER OF CHANGES
10076   034240  001367                          BNE     1$              ;BRANCH IF NOT COMPLETE
10077   034242  012602                          MOV     (SP)+,R2        ;;POP STACK INTO R2
10078   034244  012601                          MOV     (SP)+,R1        ;;POP STACK INTO R1
10079   034246  000200                          RTS     R0              ;RETURN TO MAIN PROGRAM
10080
10081
10082
10083
10084
10085
10086                                   ;THIS FILLS A BLOCK WITH INCREMENTAL DATA
10087                                   ;CALL IS
10088                                   ;       JSR     R0,@#FILL
10089                                   ;       F                       ;FROM
10090                                   ;       N                       ;NUMBER OF WORDS
10091                                   ;       S                       ;STARTING VALUE OF DATA
10092                                   ;       I                       ;INCREMENT DATA BY
10093
10094   034250                          FILL:
10095   034250  010146                          MOV     R1,-(SP)        ;;PUSH R1 ON STACK
10096   034252  010246                          MOV     R2,-(SP)        ;;PUSH R2 ON STACK
10097   034254  010346                          MOV     R3,-(SP)        ;;PUSH R3 ON STACK
10098   034256  010446                          MOV     R4,-(SP)        ;;PUSH R4 ON STACK
10099   034260  012001                          MOV     (R0)+,R1        ;R1 HAS ADDRESS WHERE DATA IS TO GO
10100   034262  012002                          MOV     (R0)+,R2        ;R2 HAS NUMBER OF WORDS TO BE FILLED
10101   034264  012003                          MOV     (R0)+,R3        ;STARTING VALUE OF DATA
10102   034266  012004                          MOV     (R0)+,R4        ;R4 HAS INCREMENT
10103
10104                                           ;*NOW DATA WILL BE FILLED
10105   034270  010321                  1$:     MOV     R3,(R1)+        ;FILL DATA
10106   034272  060403                          ADD     R4,R3           ;GET NEXT VALUE OF DATA
10107   034274  005302                          DEC     R2              ;DECREMENT COUNT
10108   034276  001374                          BNE     1$              ;BRANCH IF ALL NOT DONE
10109   034300  012604                          MOV     (SP)+,R4        ;;POP STACK INTO R4
```

```
10110  034302  012603                          MOV     (SP)+,R3        ;;POP STACK INTO R3
10111  034304  012602                          MOV     (SP)+,R2        ;;POP STACK INTO R2
10112  034306  012601                          MOV     (SP)+,R1        ;;POP STACK INTO R1
10113  034310  000200                          RTS     R0              ;RETURN TO MAIN PROGRAM
10114
10115
10116
10117
10118
10119
10120                                  ;THIS IS A SUBROUTINE TO COMPARE REGISTERS
10121                                  ;GOOD DATA IS ALREADY SAVED IN 'SAVERE'
10122                                  ;TEST DATA IS IN THE REGISTERS
10123                                  ;CALL IS
10124                                  ;       JSR     R0,@#COMREG
10125                                  ;       SAVERE                  ;GOOD DATA
10126                                  ;       RHCS1                   ;ADDRESS OF ADDRESS TEST DATA
10127                                  ;       N.                      ;RETURN FOR ERROR
10128                                  ;       RG                      ;RETURN FOR GOOD COMPARISON
10129                                  ;ON RETURN WITH ERROR '$GDDAT' HAS GOOD DATA, '$BDDAT' HAS BAD DATA
10130                                  ;'REGADR' HAS REGISTER ADDRESS
10131
10132  034312                          COMREG:
10133  034312  010146                          MOV     R1,-(SP)        ;;PUSH R1 ON STACK
10134  034314  010246                          MOV     R2,-(SP)        ;;PUSH R2 ON STACK
10135  034316  010346                          MOV     R3,-(SP)        ;;PUSH R3 ON STACK
10136  034320  010446                          MOV     R4,-(SP)        ;;PUSH R4 ON STACK
10137  034322  010546                          MOV     R5,-(SP)        ;;PUSH R5 ON STACK
10138  034324  012001                          MOV     (R0)+,R1        ;R1 HAS ADDRESS OF GOOD DATA
10139  034326  012002                          MOV     (R0)+,R2        ;R2 HAS ADDRESS OF ADDRESS OF TEST DATA
10140  034330  012003                          MOV     (R0)+,R3        ;R3 HAS NUMBER OF WORDS
10141  034332  012004                          MOV     (R0)+,R4        ;R4 HAS RETURN FOR ERROR
10142  034334  011000                          MOV     (R0),R0         ;R0 HAS RETURN ON NO ERROR
10143                                          ;*NOW SAVE REGISTERS
10144  034336  004737  035236                  JSR     PC,@#PUTREG     ;SAVE REGISTERS
10145  034342  113737  004537  002301          MOVB    @#SAVERE+25,@#AS+1;MAKE UPPER BYTE OF RHAS SAME
10146  034350  012705  177776                  MOV     #-2,R5          ;PRESET R5 TO -2
10147                                          ;*NOW COMPARES WILL MADE
10148  034354  062705  000002          1$:     ADD     #2,R5           ;INCREMENT TO INDEX
10149  034360  022122                          CMP     (R1)+,(R2)+     ;COMPARE REGISTER CONTENTS
10150  034362  001420                          BEQ     2$              ;BRANCH IF GOOD
10151  034364  014137  001124                  MOV     -(R1),@#$GDDAT  ;SAVE GOOD DATA
10152  034370  014237  001126                  MOV     -(R2),@#$BDDAT  ;SAVE BAD DATA
10153  034374  016537  002172  004500          MOV     RHWC(R5),@#REGADR       ;SAVE ADDRESS OF FAILING REGISTER
10154  034402  004714                          JSR     PC,@R4          ;RETURN TO MAIN PROGRAM
10155                                                                  ;TO PRINT ERROR
10156  034404  022122                          CMP     (R1)+,(R2)+     ;UNDO -(R1) AND -(R2) FOR ERRORS
10157  034406  017746  144526                  MOV     @SWR,-(SP)      ;GET SWITCH SETTING
10158  034412  042716  177177                  BIC     #^C600,(SP)     ;KEEP ONLY SWITCH 7 AND 8
10159  034416  022726  000200                  CMP     #SW07,(SP)+     ;IS 7 SET AND 8 DOWN
10160  034422  001402                          BEQ     3$              ;BRANCH OUT IF YES
10161  034424  005303                  2$:     DEC     R3              ;ARE ALL COMPARES DONE
10162  034426  001352                          BNE     1$              ;BRANCH IF NOT COMPLETE
10163
10164  034430                          3$:
10165  034430  012605                          MOV     (SP)+,R5        ;;POP STACK INTO R5
```

```
 10166  034432  012604                        MOV     (SP)+,R4        ;;POP STACK INTO R4
 10167  034434  012603                        MOV     (SP)+,R3        ;;POP STACK INTO R3
 10168  034436  012602                        MOV     (SP)+,R2        ;;POP STACK INTO R2
 10169  034440  012601                        MOV     (SP)+,R1        ;;POP STACK INTO R1
 10170  034442  000200                        RTS     R0              ;RETURN TO MAIN PROGRAM
 10171  034444  000000                4$:     .WORD   0               ;TEMP STORAGE
```

```
10172
10173
10174                                   ;HERE IS A DETAILED EXPLAINATION OF HOW THE LOOP ON ERROR WORKS.
10175                                   ;ON HITTING AN ERROR IF THE LOOP ON ERROR SWITCH IS SET, THE
10176                                   ;PROGRAM GOES BACK - USUALLY BACK TO THE BEGINNING OF THE TEST.
10177
10178                                   ;WHEN THIS OPERATOR SELECTABLE SCOPE LOOP IS USED THEN THE POINT
10179                                   ;THE PROGRAM GOES BACK TO CAN BE CHANGED.
10180                                   ;THE RESTRICTIONS TO THE POINT WHERE THE PROGRAM CAN GO ARE: -
10181                                   ;1.  IT MUST BE WITHIN THE TEST UNDER CONSIDERATION
10182                                   ;2.  LOOP ON ERROR SWITCH MUST BE SET
10183                                   ;3.  THE ERROR MUST OCCUR WITHIN THE TEST UNDER CONSIDERATION
10184                                   ;IF THE ERROR DOES NOT OCCUR WITHIN THE TEST UNDER CONSIDERATION
10185                                   ;THE PROGRAM WILL REVERT TO NORMAL OPERATION.  HOWEVER, IF LOOP ON
10186                                   ;TEST SWITCH IS SET AND THIS OPERATOR SELECTABLE SCOPE LOOP IS USED
10187                                   ;THEN THE PROGRAM WILL LOOP BACK TO THE SELECTED POINT WHEN IT
10188                                   ;COMES TO THE END OF THE TEST UNDER CONSIDERATION.
10189                                   ;
10190                                   ;AFTER LOOPING FOR SOME TIME IF THE LOOP SWITCH IS PUT DOWN THEN
10191                                   ;NORMAL OPERATION WILL CONTINUE.
10192
10193  034446  000000                   TESTAD: 0                       ;FIRST ADDRESS OF TEST
10194  034450                          OPERSEL:
10195  034450  005037  177776                   CLR     PS              ;MAKE PROCESSOR STATUS ZERO
10196  034454  012737  177777  041170           MOV     #-1,@#PRITEM    ;CLEAR PREVIOUS ITEM NUMBER
10197  034462  104401  034470                   TYPE    ,65$            ;;TYPE ASCIZ STRING
10198  034466  000421                           BR      64$             ;;GET OVER THE ASCIZ
10199                                  ;;65$:   .ASCIZ  <15><12>/THE PROGRAM WAS IN TEST NUMBER /
10200  034532                          64$:
10201  034532  013746  004504                   MOV     @#TSTNM,-(SP)   ;GET READY TO TYPE TEST
10202  034536  104402                           TYPOC                   ;NUMBER
10203  034540  104401  034546                   TYPE    ,67$            ;;TYPE ASCIZ STRING
10204  034544  000414                           BR      66$             ;;GET OVER THE ASCIZ
10205                                  ;;67$:   .ASCIZ  <15><12>/THE LOOP BACK PC WAS /
10206  034576                          66$:
10207  034576  013746  001110                   MOV     @#$LPERR,-(SP)  ;GET READY TO TYPE LOOP BACK PC
10208  034602  104402                           TYPOC
10209  034604  104401  001223                   TYPE    ,$CRLF
10210  034610  104401  034616                   TYPE    ,69$            ;;TYPE ASCIZ STRING
10211  034614  000430                           BR      68$             ;;GET OVER THE ASCIZ
10212                                  ;;69$:   .ASCIZ  <15><12>/SET SWITCH FOR LOOP ON ERROR OR LOOP ON TEST/
10213  034676                          68$:
10214  034676  104401  034704                   TYPE    ,71$            ;;TYPE ASCIZ STRING
10215  034702  000420                           BR      70$             ;;GET OVER THE ASCIZ
10216                                  ;;71$:   .ASCIZ  <15><12>/TYPE THE FIRST PC OF THE TEST/
10217  034744                          70$:
10218  034744  104401  034752                   TYPE    ,73$            ;;TYPE ASCIZ STRING
10219  034750  000432                           BR      72$             ;;GET OVER THE ASCIZ
10220                                  ;;73$:   .ASCIZ  <15><12>/TO BE LOOPED ON FOLLOWED BY A CARRIAGE RETURN /<15><12>
10221  035036                          72$:
10222  035036  104412                           RDOCT
10223  035040  062716  000002                   ADD     #2,(SP)         ;GET LPADR
10224  035044  012637  001106                   MOV     (SP)+,@#$LPADR
10225  035050  104401  035056                   TYPE    ,75$            ;;TYPE ASCIZ STRING
10226  035054  000417                           BR      74$             ;;GET OVER THE ASCIZ
10227                                  ;;75$:   .ASCIZ  <15><12>/TYPE THE PC WHERE YOU WANT/
```

```
10228  035114                          74$:
10229  035114  104401  035122                  TYPE    ,77$             ;;TYPE ASCIZ STRING
10230  035120  000440                          BR      76$              ;;GET OVER THE ASCIZ
10231                                  ;;77$:   .ASCIZ  <15><12>/ THE PROGRAM TO LOOP BACK TO FOLLOWED BY A CARRIAGE RETURN /<15
10232  035222                          76$:
10233  035222  104412                          RDOCT
10234  035224  012637  001110                  MOV     (SP)+,@#$LPERR  ;GET LPERR
10235  035230  013746  001106                  MOV     @#$LPADR,-(SP)
10236  035234  000002                          RTI
10237
10238
10239
10240
10241
10242
10243                                  ;*THIS SAVES THE CONTENTS OF ALL HARDWARE REGISTERS
10244                                  ;*IN MEMORY LOCATIONS TAGED FROM "WC" TO "EC2"
10245                                  ;*THIS IS DONE SO THAT COMPARES ARE DONE WITH SAVED LOCATIONS
10246                                  ;*AND NOT THE REGISTERS THEMSELVES.   THIS WILL MAKE
10247                                  ;*ERROR PRINTOUTS FOR GOOD AND BAD DATA ALWAYS DIFFRENT
10248
10249
10250
10251
10252  035236                          PUTREG:
10253  035236  010046                          MOV     R0,-(SP)        ;;PUSH R0 ON STACK
10254  035240  010146                          MOV     R1,-(SP)        ;;PUSH R1 ON STACK
10255  035242  010246                          MOV     R2,-(SP)        ;;PUSH R2 ON STACK
10256  035244  012700  002172                  MOV     #RHWC,R0        ;STARTING ADDRESS OF REGISTERS
10257  035250  012701  002254                  MOV     #WC,R1          ;STARTING ADDRESS OF SAVING LOCATIONS
10258  035254  012702  000022                  MOV     #RHCC-RHWC+2/2,R2 ;NUMBER OF REG. INTO R2
10259  035260  013021                  10$:    MOV     @(R0)+,(R1)+    ;SAVE HARDWARE REG.
10260  035262  005302                          DEC     R2
10261  035264  001375                          BNE     10$
10262  035266  012602                          MOV     (SP)+,R2        ;;POP STACK INTO R2
10263  035270  012601                          MOV     (SP)+,R1        ;;POP STACK INTO R1
10264  035272  012600                          MOV     (SP)+,R0        ;;POP STACK INTO R0
10265  035274  000207                          RTS     PC
```

```
10266                                   ;THIS IS A DATA  COMMAND SETUP SUBROUTINE
10267                                   ;THE CALL IS
10268                                   ;       JSR     R0,@#RUN
10269                                   ;       C                       ;CYLINDER
10270                                   ;.BYTE  S                       ;SECTOR
10271                                   ;.BYTE  T                       ;TRACK
10272                                   ;       -W                      ;WORD COUNT
10273                                   ;       B                       ;BUS ADDRESS
10274                                   ;       BAI                     ;BUS ADDRESS INHIBIT
10275                                   ;       FMT22!ECI!HCI           ;FMT22=1 =16 BIT WORDS
10276                                   ;                               ;ECI = ECC CORRECTION INHIBIT
10277                                   ;                               ;HCI = HEADER COMPARE INHIBIT
10278                                   ;       COM                     ;COMMAND ADDRESS
10279  035276  012077  144710          RUN:    MOV     (R0)+,@RHCA     ;CYLINDER
10280  035302  012077  144676                  MOV     (R0)+,@RHDST    ;DESIRED SECTOR/TRACK
10281  035306  012077  144660                  MOV     (R0)+,@RHWC     ;WORD COUNT
10282  035312  012077  144656                  MOV     (R0)+,@RHBA     ;BUS ADDRESS
10283  035316  013746  004616                  MOV     @#UNIT,-(SP)    ;GET UNIT NO
10284  035322  052016                          BIS     (R0)+,(SP)      ;SET BUS ADDRESS INHIBIT
10285  035324  012677  144646                  MOV     (SP)+,@RHCS2    ;UNIT NO AND BAI TO RHCS2
10286  035330  012077  144654                  MOV     (R0)+,@RHOF     ;FORMAT, ECC INHIBIT, HEADER
10287                                                                  ;COMPARE, IF THERE
10288  035334  013077  144640                  MOV     @(R0)+,@RHCS1   ;COMMAND IN RHCS1
10289  035340  000200                          RTS     R0              ;RETURN TO MAIN PROGRAM
10290
10291
10292
10293
10294                                   ;THIS IS A SUBROUTINE TO COMPARE TWO BLOCKS IN MEMORY
10295                                   ;R1 HAS GOOD DATA BUFFER ADDRESS
10296                                   ;R2 HAS TEST DATA BUFFER ADDRESS
10297                                   ;R5 HAS ADDRESS OF RETURN ON ERROR
10298                                   ;R3 HAS NUMBER OF WORDS TO BE COMPARED
10299                                   ;R4 HAS ONE MORE THAN NUMBER OF WORDS TO BE COMPARED
10300                                   ;CALL IS
10301                                   ;       JSR     R0,@#COMPAR
10302                                   ;       G               ;ADDRESS OF GOOD DATA
10303                                   ;       T               ;ADDRESS OF TEST DATA
10304                                   ;       N               ;NUMBER OF WORDS TO BE COMPARED
10305                                   ;       RE              ;RETURN ON ERROR
10306                                   ;       RG              ;RETURN ON NO ERROR
10307
10308
10309
10310  035342                           COMPAR:
10311  035342  010146                          MOV     R1,-(SP)        ;;PUSH R1 ON STACK
10312  035344  010246                          MOV     R2,-(SP)        ;;PUSH R2 ON STACK
10313  035346  010346                          MOV     R3,-(SP)        ;;PUSH R3 ON STACK
10314  035350  010446                          MOV     R4,-(SP)        ;;PUSH R4 ON STACK
10315  035352  010546                          MOV     R5,-(SP)        ;;PUSH R5 ON STACK
10316  035354  012001                          MOV     (R0)+,R1        ;ADDRESS OF GOOD DATA BUFFER
10317  035356  012002                          MOV     (R0)+,R2        ;ADDRESS OF TEST DATA BUFFER
10318  035360  012003                          MOV     (R0)+,R3        ;NO OF WORDS TO BE COMPARED
10319  035362  012005                          MOV     (R0)+,R5        ;RETURN ON ERROR
10320  035364  011000                          MOV     (R0),R0         ;RETURN ON NO ERROR
10321  035366  010304                          MOV     R3,R4           ;NO OF WORDS TO BE COMPARED
```

```
10322  035370  005204                          INC     R4
10323  035372  010437  004502          1$:     MOV     R4,@#ERWORD     ;FOR ERROR WORD NO
10324  035376  022122                          CMP     (R1)+,(R2)+     ;COMPARE GOOD WITH TEST DATA
10325  035400  001417                          BEQ     2$              ;BRANCH IF GOOD
10326
10327  035402  014137  001124                  MOV     -(R1),@#$GDDAT  ;GOOD DATA
10328  035406  014237  001126                  MOV     -(R2),@#$BDDAT  ;BAD DATA
10329  035412  160337  004502                  SUB     R3,@#ERWORD     ;ERROR WORD NO.
10330  035416  004715                          JSR     PC,@R5  ;RETURN TO PRINT ERROR
10331  035420  022122                          CMP     (R1)+,(R2)+     ;UNDO -(R1) AND -(R2) FOR ERRORS
10332  035422  017746  143512                  MOV     @SWR,-(SP)      ;GET SWITCH SETTING
10333  035426  042716  177177                  BIC     #^C600,(SP)     ;KEEP ONLY SWITCH 7 AND 8
10334  035432  022726  000200                  CMP     #SW07,(SP)+     ;IS 7 SET AND 8 RESET
10335  035436  001402                          BEQ     3$              ;BRANCH OUT IF YES
10336  035440  005303                  2$:     DEC     R3              ;COUNT
10337  035442  001353                          BNE     1$              ;BRANCH IF ALL NOT DEVICE
10338  035444                          3$:
10339  035444  012605                          MOV     (SP)+,R5        ;;POP STACK INTO R5
10340  035446  012604                          MOV     (SP)+,R4        ;;POP STACK INTO R4
10341  035450  012603                          MOV     (SP)+,R3        ;;POP STACK INTO R3
10342  035452  012602                          MOV     (SP)+,R2        ;;POP STACK INTO R2
10343  035454  012601                          MOV     (SP)+,R1        ;;POP STACK INTO R1
10344  035456  000200                          RTS     R0              ;RETURN TO MAIN PROGRAM
10345                                  ;*      THIS ROUTINE WILL ALLOW THE CHANGE OF THE BASE
10346                                  ;*      ADDRESS FROM 176700 TO ANY TYPED VALUE
10347
10348  035460                          BASECH:
10349  035460  104401  035466                  TYPE    ,65$            ;;TYPE ASCIZ STRING
10350  035464  000425                          BR      64$             ;;GET OVER THE ASCIZ
10351                                  ;;65$:  .ASCIZ  <15><12>/PRESENT BASE ADDRESS OF REGISTERS IS   /
10352  035540                          64$:
10353  035540  013746  002200                  MOV     @#RHCS1,-(SP)   ;GET READY TO TYPE OLD BASE
10354  035544  104402                          TYPOC
10355  035546  104401  035554                  TYPE    ,67$            ;;TYPE ASCIZ STRING
10356  035552  000425                          BR      66$             ;;GET OVER THE ASCIZ
10357                                  ;;67$:  .ASCIZ  <15><12>/TYPE NEW BASE ADDRESS FOLLOWED BY 'CR' /
10358  035626                          66$:
10359  035626  004737  037562                  JSR     PC,@#$TKINT     ;INITIALIZE THE TTY KEYBOARD
10360  035632  104412                          RDOCT
10361  035634  012700  002170                  MOV     #RHDB,R0        ;GET STARTING ADDRESS OF REGISTERS
10362  035640  012701  000026                  MOV     #22.,R1         ;NUMBER OF REGISTERS
10363  035644  012737  036450  000004          MOV     #ADTIMO,@#4     ;SET UP TRAP CATCHER FOR TEST
10364  035652  021637  002200                  CMP     @SP,@#RHCS1     ;NEW ADDRESS
10365  035656  001407                          BEQ     1$              ;NO, JUST OLD ONE RETYPED
10366  035660  005776  000000                  TST     @0(SP)          ;DO THE ADDRESS ACCESS
10367  035664  163716  002200                  SUB     @#RHCS1,@SP     ;GET THE ADDRESS OFFSET
10368  035670  061620                  2$:     ADD     @SP,(R0)+       ;AND PLUG IT IN
10369  035672  005301                          DEC     R1              ;ONE LESS REGISTER TO DO
10370  035674  001375                          BNE     2$              ;BUT WE'RE NOT DONE YET!
10371  035676                          1$:
10372  035676  104401  035704                  TYPE    ,69$            ;;TYPE ASCIZ STRING
10373  035702  000417                          BR      68$             ;;GET OVER THE ASCIZ
10374                                  ;;69$:  .ASCIZ  <15><12>/PRESENT VECTOR ADDRESS IS /
10375  035742                          68$:
10376  035742  013746  002166                  MOV     @#RPVEC,-(SP)   ;GET READY TO TYPE OLD VECTOR ADDRESS
10377  035746  104402                          TYPOC
```

```
10378  035750  104401  035756                  TYPE    ,71$            ;;TYPE ASCIZ STRING
10379  035754  000437                          BR      70$             ;;GET OVER THE ASCIZ
10380                                  ;;71$:   .ASCIZ  <15><12>/TYPE NEW VECTOR ADDRESS OR RETYPE OLD ONE FOLLOWED BY "CR" /
10381  036054                          70$:
10382  036054  104412                          RDOCT
10383  036056  012637  002166                  MOV     (SP)+,@#RPVEC   ;SETUP VECTOR ADDRESS
10384  036062  104401  036070                  TYPE    ,73$            ;;TYPE ASCIZ STRING
10385  036066  000417                          BR      72$             ;;GET OVER THE ASCIZ
10386                                  ;;73$:   .ASCIZ  <15><12>/NEW BASE WILL REMAIN    - /
10387  036126                          72$:
10388  036126  013746  002200                  MOV     @#RHCS1,-(SP)
10389  036132  104402                          TYPOC
10390  036134  104401  036142                  TYPE    ,75$            ;;TYPE ASCIZ STRING
10391  036140  000417                          BR      74$             ;;GET OVER THE ASCIZ
10392                                  ;;75$:   .ASCIZ  <15><12>/NEW VECTOR WILL REMAIN  - /
10393  036200                          74$:
10394  036200  013746  002166                  MOV     @#RPVEC,-(SP)
10395  036204  104402                          TYPOC
10396  036206  104401  036214                  TYPE    ,77$            ;;TYPE ASCIZ STRING
10397  036212  000417                          BR      76$             ;;GET OVER THE ASCIZ
10398                                  ;;77$:   .ASCIZ  <15><12>/UNTIL PROGRAM IS RELOADED./
10399  036252                          76$:
10400  036252  104401  036260                  TYPE    ,79$            ;;TYPE ASCIZ STRING
10401  036256  000402                          BR      78$             ;;GET OVER THE ASCIZ
10402                                  ;;79$:   .ASCIZ  <15><12>/ /
10403  036264                          78$:
10404  036264  104401  036272                  TYPE    ,81$            ;;TYPE ASCIZ STRING
10405  036270  000424                          BR      80$             ;;GET OVER THE ASCIZ
10406                                  ;;81$:   .ASCIZ  <15><12>/UNLESS HALTED AND MANUALLY RESTARTED,/
10407  036342                          80$:
10408  036342  104401  036350                  TYPE    ,83$            ;;TYPE ASCIZ STRING
10409  036346  000426                          BR      82$             ;;GET OVER THE ASCIZ
10410                                  ;;83$:   .ASCIZ  <15><12>/PROGRAM WILL AUTOMATICALLY RESTART FROM /
10411  036424                          82$:
10412  036424  012746  000200                  MOV     #RA,-(SP)
10413  036430  104402                          TYPOC
10414  036432  104401  036440                  TYPE    ,85$            ;;TYPE ASCIZ STRING
10415  036436  000402                          BR      84$             ;;GET OVER THE ASCIZ
10416                                  ;;85$:   .ASCIZ  <15><12>/ /
10417  036444                          84$:
10418  036444  000137  004712                  JMP     @#BEGIN         ;DO IT OVER AGAIN
10419  036450                          ADTIMO:
10420  036450  104401  036456                  TYPE    ,65$            ;;TYPE ASCIZ STRING
10421  036454  000424                          BR      64$             ;;GET OVER THE ASCIZ
10422                                  ;;65$:   .ASCIZ  <15><12><377>/SELECTED ADDRESS DID NOT RESPOND.   /
10423  036526                          64$:
10424  036526  022626                          CMP     (SP)+,(SP)+     ;RESTORE STACK
10425  036530  000137  035460                  JMP     @#BASECH        ;AND DO THE QUERY AGAIN!
10426
```

```
10427                                   ;;*****************************************************************
10428  036534                           RPVECT:
10429  036534  104401  036542                   TYPE    ,65$            ;;TYPE ASCIZ STRING
10430  036540  000424                           BR      64$             ;;GET OVER THE ASCIZ
10431                                   ;;65$:  .ASCIZ  /UNEXPECTED INTERRUPT FROM RP04 @ PC = /
10432  036612                           64$:
10433  036612  104402                           TYPOC                           ;TYPE FROM PC
10434  036614  012777  036534  143344         MOV     #RPVECT,@RPVEC          ;RESTORE TRAP RP04 VECTOR
10435  036622  000000                           HALT                            ;CHANGE TO CONTINUE
10436                                   ;;*****************************************************************
10437
10438
```

```
10439                                           .SBTTL  SYSMAC LIBRARY ROUTINES
10440
10441
10442                                   .SBTTL  SCOPE HANDLER ROUTINE
10443
10444                                   ;;*********************************************************************
10445                                   ;*THIS ROUTINE CONTROLS THE LOOPING OF SUBTESTS. IT WILL INCREMENT
10446                                   ;*AND LOAD THE TEST NUMBER($TSTNM) INTO THE DISPLAY REG.(DISPLAY<7:0>)
10447                                   ;*AND LOAD THE ERROR FLAG ($ERFLG) INTO DISPLAY<15:08>
10448                                   ;*THE SWITCH OPTIONS PROVIDED BY THIS ROUTINE ARE:
10449                                   ;*SW14=1         LOOP ON TEST
10450                                   ;*SW11=1         INHIBIT ITERATIONS
10451                                   ;*SW09=1         LOOP ON ERROR
10452                                   ;*SW08=1         LOOP ON TEST IN SWR<7:0>
10453                                   ;*CALL
10454                                   ;*      SCOPE            ;;SCOPE=IOT
10455
10456  036624                           $SCOPE:
10457  036624  104407                          CKSWR                   ;;TEST  FOR CHANGE IN SOFT-SWR
10458  036626  032777  040000  142304  1$:     BIT     #BIT14,@SWR     ;;LOOP ON PRESENT TEST?
10459  036634  001111                          BNE     $OVER           ;;YES IF SW14=1
10460                                   ;#####START OF CODE FOR THE XOR TESTER#####
10461  036636  000416                   $XTSTR: BR      6$              ;;IF RUNNING ON THE "XOR" TESTER CHANGE
10462                                                                   ;;THIS INSTRUCTION TO A "NOP" (NOP=240)
10463  036640  013746  000004                  MOV     @#ERRVEC,-(SP)  ;;SAVE THE CONTENTS OF THE ERROR VECTOR
10464  036644  012737  036664  000004          MOV     #5$,@#ERRVEC    ;;SET FOR TIMEOUT
10465  036652  005737  177060                  TST     @#177060        ;;TIME OUT ON XOR?
10466  036656  012637  000004                  MOV     (SP)+,@#ERRVEC  ;;RESTORE THE ERROR VECTOR
10467  036662  000463                          BR      $SVLAD          ;;GO TO THE NEXT TEST
10468  036664  022626                   5$:     CMP     (SP)+,(SP)+     ;;CLEAR THE STACK AFTER A TIME OUT
10469  036666  012637  000004                  MOV     (SP)+,@#ERRVEC  ;;RESTORE THE ERROR VECTOR
10470  036672  000423                          BR      7$              ;;LOOP ON THE PRESENT TEST
10471  036674                           6$:;#####END OF CODE FOR THE XOR TESTER#####
10472  036674  032777  000400  142236          BIT     #BIT08,@SWR     ;;LOOP ON SPEC. TEST?
10473  036702  001404                          BEQ     2$              ;;BR IF NO
10474  036704  127737  142230  001102          CMPB    @SWR,$TSTNM     ;;ON THE RIGHT TEST?    SWR<7:0>
10475  036712  001462                          BEQ     $OVER           ;;BR IF YES
10476  036714  105737  001103           2$:     TSTB    $ERFLG          ;;HAS AN ERROR OCCURRED?
10477  036720  001421                          BEQ     3$              ;;BR IF NO
10478  036722  123737  001115  001103          CMPB    $ERMAX,$ERFLG   ;;MAX. ERRORS FOR THIS TEST OCCURRED?
10479  036730  101015                          BHI     3$              ;;BR IF NO
10480  036732  032777  001000  142200          BIT     #BIT09,@SWR     ;;LOOP ON ERROR?
10481  036740  001404                          BEQ     4$              ;;BR IF NO
10482  036742  013737  001110  001106  7$:     MOV     $LPERR,$LPADR   ;;SET LOOP ADDRESS TO LAST SCOPE
10483  036750  000443                          BR      $OVER
10484  036752  105037  001103           4$:     CLRB    $ERFLG          ;;ZERO THE ERROR FLAG
10485  036756  005037  001212                  CLR     $TIMES          ;;CLEAR THE NUMBER OF ITERATIONS TO MAKE
10486  036762  000415                          BR      1$              ;;ESCAPE TO THE NEXT TEST
10487  036764  032777  004000  142146  3$:     BIT     #BIT11,@SWR     ;;INHIBIT ITERATIONS?
10488  036772  001011                          BNE     1$              ;;BR IF YES
10489  036774  005737  001100                  TST     $PASS           ;;IF FIRST PASS OF PROGRAM
10490  037000  001406                          BEQ     1$              ;;      INHIBIT ITERATIONS
10491  037002  005237  001104                  INC     $ICNT           ;;INCREMENT ITERATION COUNT
10492  037006  023737  001212  001104          CMP     $TIMES,$ICNT    ;;CHECK THE NUMBER OF ITERATIONS MADE
10493  037014  002021                          BGE     $OVER           ;;BR IF MORE ITERATION REQUIRED
10494  037016  012737  000001  001104  1$:     MOV     #1,$ICNT        ;;REINITIALIZE THE ITERATION COUNTER
```

```
10495  037024  013737  037074  001212          MOV     $MXCNT,$TIMES   ;;SET NUMBER OF ITERATIONS TO DO
10496  037032  105237  001102          $SVLAD: INCB    $TSTNM          ;;COUNT TEST NUMBERS
10497  037036  011637  001106                  MOV     (SP),$LPADR     ;;SAVE SCOPE LOOP ADDRESS
10498  037042  011637  001110                  MOV     (SP),$LPERR     ;;SAVE ERROR LOOP ADDRESS
10499  037046  005037  001214                  CLR     $ESCAPE         ;;CLEAR THE ESCAPE FROM ERROR ADDRESS
10500  037052  112737  000001  001115          MOVB    #1,$ERMAX       ;;ONLY ALLOW ONE(1) ERROR ON NEXT TEST
10501  037060  013777  001102  142054  $OVER:  MOV     $TSTNM,@DISPLAY ;;DISPLAY TEST NUMBER
10502  037066  013716  001106                  MOV     $LPADR,(SP)     ;;FUDGE RETURN ADDRESS
10503  037072  000002                          RTI                     ;;FIXES PS
10504  037074  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

037076                          $TYPDS:
037076  010046                          MOV     R0,-(SP)        ;;PUSH R0 ON STACK
037100  010146                          MOV     R1,-(SP)        ;;PUSH R1 ON STACK
037102  010246                          MOV     R2,-(SP)        ;;PUSH R2 ON STACK
037104  010346                          MOV     R3,-(SP)        ;;PUSH R3 ON STACK
037106  010546                          MOV     R5,-(SP)        ;;PUSH R5 ON STACK
037110  012746  020200                  MOV     #20200,-(SP)    ;;SET BLANK SWITCH AND SIGN
037114  016605  000020                  MOV     20(SP),R5       ;;GET THE INPUT NUMBER
037120  100004                          BPL     1$              ;;BR IF INPUT IS POS.
037122  005405                          NEG     R5              ;;MAKE THE BINARY NUMBER POS.
037124  112766  000055  000001          MOVB    #'-,1(SP)       ;;MAKE THE ASCII NUMBER NEG.
037132  005000                  1$:     CLR     R0              ;;ZERO THE CONSTANTS INDEX
037134  012703  037312                  MOV     #$DBLK,R3       ;;SETUP THE OUTPUT POINTER
037140  112723  000040                  MOVB    #' ,(R3)+       ;;SET THE FIRST CHARACTER TO A BLANK
037144  005002                  2$:     CLR     R2              ;;CLEAR THE BCD NUMBER
037146  016001  037302                  MOV     $DTBL(R0),R1    ;;GET THE CONSTANT
037152  160105                  3$:     SUB     R1,R5           ;;FORM THIS BCD DIGIT
037154  002402                          BLT     4$              ;;BR IF DONE
037156  005202                          INC     R2              ;;INCREASE THE BCD DIGIT BY 1
037160  000774                          BR      3$
037162  060105                  4$:     ADD     R1,R5           ;;ADD BACK THE CONSTANT
037164  005702                          TST     R2              ;;CHECK IF BCD DIGIT=0
037166  001002                          BNE     5$              ;;FALL THROUGH IF 0
037170  105716                          TSTB    (SP)            ;;STILL DOING LEADING 0'S?
037172  100407                          BMI     7$              ;;BR IF YES
037174  106316                  5$:     ASLB    (SP)            ;;MSD?
037176  103003                          BCC     6$              ;;BR IF NO
037200  116663  000001  177777          MOVB    1(SP),-1(R3)    ;;YES--SET THE SIGN
037206  052702  000060          6$:     BIS     #'0,R2          ;;MAKE THE BCD DIGIT ASCII
037212  052702  000040          7$:     BIS     #' ,R2          ;;MAKE IT A SPACE IF NOT ALREADY A DIGIT
037216  110223                          MOVB    R2,(R3)+        ;;PUT THIS CHARACTER IN THE OUTPUT BUFFER
037220  005720                          TST     (R0)+           ;;JUST INCREMENTING
037222  020027  000010                  CMP     R0,#10          ;;CHECK THE TABLE INDEX
037226  002746                          BLT     2$              ;;GO DO THE NEXT DIGIT
037230  003002                          BGT     8$              ;;GO TO EXIT
037232  010502                          MOV     R5,R2           ;;GET THE LSD
037234  000764                          BR      6$              ;;GO CHANGE TO ASCII
037236  105726                  8$:     TSTB    (SP)+           ;;WAS THE LSD THE FIRST NON-ZERO?
037240  100003                          BPL     9$              ;;BR IF NO
037242  116663  177777  177776          MOVB    -1(SP),-2(R3)   ;;YES--SET THE SIGN FOR TYPING
037250  105013                  9$:     CLRB    (R3)            ;;SET THE TERMINATOR
037252  012605                          MOV     (SP)+,R5        ;;POP STACK INTO R5
037254  012603                          MOV     (SP)+,R3        ;;POP STACK INTO R3
037256  012602                          MOV     (SP)+,R2        ;;POP STACK INTO R2
```

```
10561  037260  012601                          MOV     (SP)+,R1        ;;POP STACK INTO R1
10562  037262  012600                          MOV     (SP)+,R0        ;;POP STACK INTO R0
10563  037264  104401  037312                  TYPE    ,$DBLK          ;;NOW TYPE THE NUMBER
10564  037270  016666  000002  000004          MOV     2(SP),4(SP)     ;;ADJUST THE STACK
10565  037276  012616                          MOV     (SP)+,(SP)
10566  037300  000002                          RTI                     ;;RETURN TO USER
10567  037302  023420                $DTBL:    10000.
10568  037304  001750                          1000.
10569  037306  000144                          100.
10570  037310  000012                          10.
10571  037312  000004                $DBLK:    .BLKW   4
```

```
10572                                   .SBTTL  TYPE ROUTINE
10573
10574                                   ;;*****************************************************************
10575                                   ;*ROUTINE TO TYPE ASCIZ MESSAGE. MESSAGE MUST TERMINATE WITH A 0 BYTE.
10576                                   ;*THE ROUTINE WILL INSERT A NUMBER OF NULL CHARACTERS AFTER A LINE FEED.
10577                                   ;*NOTE1:        $NULL CONTAINS THE CHARACTER TO BE USED AS THE FILLER CHARACTER.
10578                                   ;*NOTE2:        $FILLS CONTAINS THE NUMBER OF FILLER CHARACTERS REQUIRED.
10579                                   ;*NOTE3:        $FILLC CONTAINS THE CHARACTER TO FILL AFTER.
10580                                   ;*
10581                                   ;*CALL:
10582                                   ;*1) USING A TRAP INSTRUCTION
10583                                   ;*      TYPE    ,MESADR         ;;MESADR IS FIRST ADDRESS OF AN ASCIZ STRING
10584                                   ;*OR
10585                                   ;*      TYPE
10586                                   ;*      MESADR
10587                                   ;*
10588
10589  037322  105737  001157          $TYPE:  TSTB    $TPFLG          ;;IS THERE A TERMINAL?
10590  037326  100002                          BPL     1$              ;;BR IF YES
10591  037330  000000                          HALT                    ;;HALT HERE IF NO TERMINAL
10592  037332  000407                          BR      3$              ;;LEAVE
10593  037334  010046                  1$:     MOV     R0,-(SP)        ;;SAVE R0
10594  037336  017600  000002                  MOV     @2(SP),R0       ;;GET ADDRESS OF ASCIZ STRING
10595  037342  112046                  2$:     MOVB    (R0)+,-(SP)     ;;PUSH CHARACTER TO BE TYPED ONTO STACK
10596  037344  001005                          BNE     4$              ;;BR IF IT ISN'T THE TERMINATOR
10597  037346  005726                          TST     (SP)+           ;;IF TERMINATOR POP IT OFF THE STACK
10598  037350  012600                  60$:    MOV     (SP)+,R0        ;;RESTORE R0
10599  037352  062716  000002          3$:     ADD     #2,(SP)         ;;ADJUST RETURN PC
10600  037356  000002                          RTI                     ;;RETURN
10601  037360  122716  000011          4$:     CMPB    #HT,(SP)        ;;BRANCH IF <HT>
10602  037364  001430                          BEQ     8$
10603  037366  122716  000200                  CMPB    #CRLF,(SP)      ;;BRANCH IF NOT <CRLF>
10604  037372  001006                          BNE     5$
10605  037374  005726                          TST     (SP)+           ;;POP  <CR><LF> EQUIV
10606  037376  104401                          TYPE                    ;;TYPE A CR AND LF
10607  037400  001223                          $CRLF
10608  037402  105037  037536                  CLRB    $CHARCNT        ;;CLEAR CHARACTER COUNT
10609  037406  000755                          BR      2$              ;;GET NEXT CHARACTER
10610  037410  004737  037472          5$:     JSR     PC,$TYPEC       ;;GO TYPE THIS CHARACTER
10611  037414  123726  001156          6$:     CMPB    $FILLC,(SP)+    ;;IS IT TIME FOR FILLER CHARS.?
10612  037420  001350                          BNE     2$              ;;IF NO GO GET NEXT CHAR.
10613  037422  013746  001154                  MOV     $NULL,-(SP)     ;;GET # OF FILLER CHARS. NEEDED
10614                                                                  ;;AND THE NULL CHAR.
10615  037426  105366  000001          7$:     DECB    1(SP)           ;;DOES A NULL NEED TO BE TYPED?
10616  037432  002770                          BLT     6$              ;;BR IF NO--GO POP THE NULL OFF OF STACK
10617  037434  004737  037472                  JSR     PC,$TYPEC       ;;GO TYPE A NULL
10618  037440  105337  037536                  DECB    $CHARCNT        ;;DO NOT COUNT AS A COUNT
10619  037444  000770                          BR      7$              ;;LOOP
10620
10621                                   ;HORIZONTAL TAB PROCESSOR
10622
10623  037446  112716  000040          8$:     MOVB    #' ,(SP)        ;;REPLACE TAB WITH SPACE
10624  037452  004737  037472          9$:     JSR     PC,$TYPEC       ;;TYPE A SPACE
10625  037456  132737  000007  037536          BITB    #7,$CHARCNT     ;;BRANCH IF NOT AT
10626  037464  001372                          BNE     9$              ;;TAB STOP
10627  037466  005726                          TST     (SP)+           ;;POP SPACE OFF STACK
```

```
 10628  037470  000724                          BR      2$              ;;GET NEXT CHARACTER
 10629  037472  105777  141452          $TYPEC: TSTB    @$TPS           ;;WAIT UNTIL PRINTER IS READY
 10630  037476  100375                          BPL     $TYPEC
 10631  037500  116677  000002  141444          MOVB    2(SP),@$TPB     ;;LOAD CHAR TO BE TYPED INTO DATA REG.
 10632  037506  122766  000015  000002          CMPB    #CR,2(SP)       ;;IS CHARACTER A CARRIAGE RETURN?
 10633  037514  001003                          BNE     1$              ;;BRANCH IF NO
 10634  037516  105037  037536                  CLRB    $CHARCNT        ;;YES--CLEAR CHARACTER COUNT
 10635  037522  000406                          BR      $TYPEX          ;;EXIT
 10636  037524  122766  000012  000002  1$:     CMPB    #LF,2(SP)       ;;IS CHARACTER A LINE FEED?
 10637  037532  001402                          BEQ     $TYPEX          ;;BRANCH IF YES
 10638  037534  105227                          INCB    (PC)+           ;;COUNT THE CHARACTER
 10639  037536  000000          $CHARCNT:.WORD  0               ;;CHARACTER COUNT STORAGE
 10640  037540  000207          $TYPEX: RTS     PC
 10641
```

```
                                        .SBTTL  TTY INPUT ROUTINE

                                        ;;*********************************************************************
                                        .ENABL  LSB
037542  000000                          $TKCNT: .WORD   0               ;;NUMBER OF ITEMS IN QUEUE
037544  000000                          $TKQIN: .WORD   0               ;;INPUT POINTER
037546  000000                          $TKQOUT: .WORD  0               ;;OUTPUT POINTER
037550  000011                          $TKQSRT: .BLKB  9.              ;;TTY KEYBOARD QUEUE
        037561                          $TKQEND=.
        037562                          .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
                                        ;
037562  005037  037542                  $TKINT: CLR     $TKCNT          ;;CLEAR COUNT OF ITEMS IN QUEUE
037566  012737  037550  037544                  MOV     #$TKQSRT,$TKQIN ;;MOVE THE STARTING ADDRESS OF THE
037574  013737  037544  037546                  MOV     $TKQIN,$TKQOUT  ;;QUEUE INTO THE INPUT & OUTPUT POINTERS.
037602  012737  037632  000060                  MOV     #$TKSRV,@#TKVEC ;;INITIALIZE THE KEYBOARD VECTOR
037610  012737  000200  000062                  MOV     #200,@#TKVEC+2  ;;"BR" LEVEL 4
037616  005777  141324                        TST     @$TKB           ;;CLEAR DONE FLAG
037622  012777  000100  141314                MOV     #100,@$TKS      ;;ENABLE TTY KEYBOARD INTERRUPT
037630  000207                          RTS     PC              ;;RETURN TO CALLER

                                        ;*TK SERVICE ROUTINE
                                        ;*THIS ROUTINE WILL SERVICE THE TTY KEYBOARD INTERRUPT
                                        ;*BY READING THE CHARACTER FROM THE INPUT BUFFER AND PUTTING
                                        ;*IT IN THE QUEUE.
                                        ;*IF THE CHARACTER IS A "CONTROL-C" (^C) $TKINT IS CALLED AND
                                        ;*UPON RETURN EXIT IS MADE TO THE "CONTROL-C" RESTART ADDRESS (OPERSEL)
                                        ;
037632  117746  141310                  $TKSRV: MOVB    @$TKB,-(SP)     ;;PICKUP THE CHARACTER
037636  042716  177600                          BIC     #^C177,(SP)     ;;STRIP THE JUNK
037642  021627  000003                          CMP     (SP),#3         ;;IS IT A CONTROL C?
037646  001007                                  BNE     1$              ;;BRANCH IF NO
037650  104401  040621                          TYPE    ,$CNTLC         ;;TYPE A CONTROL-C (^C)
037654  004737  037562                          JSR     PC,$TKINT       ;;INIT THE KEYBOARD
037660  005726                                  TST     (SP)+           ;;CLEAN UP STACK
037662  000137  034450                          JMP     OPERSEL         ;;CONTROL C RESTART
037666  021627  000007                  1$:     CMP     (SP),#7         ;;IS IT A CONTROL G?
037672  001004                                  BNE     2$              ;;BRANCH IF NO
037674  022737  000176  001140                  CMP     #SWREG,SWR      ;;IS SOFT-SWR SELECTED?
037702  001500                                  BEQ     6$              ;;GO TO SWR CHANGE

037704                                  2$:
037704  022737  000011  037542                  CMP     #9.,$TKCNT      ;;IS THE QUEUE FULL?
037712  001004                                  BNE     3$              ;;BRANCH IF NO
037714  104401  001216                          TYPE    ,$BELL          ;;RING THE TTY BELL
037720  005726                                  TST     (SP)+           ;;CLEAN CHARACTER OFF OF STACK
037722  000451                                  BR      5$              ;;EXIT
037724  021627  000023                  3$:     CMP     (SP),#23        ;;IS IT A CONTROL-S?
037730  001021                                  BNE     32$             ;;BRANCH IF NO
```

```
10698  037732  005077  141206                  CLR     @$TKS           ;;DISABLE TTY KEYBOARD INTERRUPTS
10699  037736  005726                          TST     (SP)+           ;;CLEAN CHAR OFF STACK
10700  037740  105777  141200          31$:    TSTB    @$TKS           ;;WAIT FOR A CHAR
10701  037744  100375                          BPL     31$             ;;LOOP UNTIL ITS THERE
10702  037746  117746  141174                  MOVB    @$TKB,-(SP)     ;;GET THE CHARACTER
10703  037752  042716  177600                  BIC     #^C177,(SP)     ;;MAKE IT 7-BIT ASCII
10704  037756  022627  000021                  CMP     (SP)+,#21       ;;IS IT A CONTROL-Q?
10705  037762  001366                          BNE     31$             ;;BRANCH IF NO
10706  037764  012777  000100  141152          MOV     #100,@$TKS      ;;REENABLE TTY KEYBOARD INTERRUPTS
10707  037772  000002                          RTI                     ;;RETURN
10708  037774  005237  037542          32$:    INC     $TKCNT          ;;COUNT THIS CHARACTER
10709  040000  021627  000140                  CMP     (SP),#140       ;;IS IT UPPER CASE?
10710  040004  002405                          BLT     4$              ;;BRANCH IF YES
10711  040006  021627  000175                  CMP     (SP),#175       ;;IS IT A SPECIAL CHAR?
10712  040012  003002                          BGT     4$              ;;BRANCH IF YES
10713  040014  042716  000040                  BIC     #40,(SP)        ;;MAKE IT UPPER CASE
10714  040020  112677  177520          4$:     MOVB    (SP)+,@$TKQIN   ;;AND PUT IT IN QUEUE
10715  040024  005237  037544                  INC     $TKQIN          ;;UPDATE THE POINTER
10716  040030  023727  037544  037561          CMP     $TKQIN,#$TKQEND ;;GO OFF THE END?
10717  040036  001003                          BNE     5$              ;;BRANCH IF NO
10718  040040  012737  037550  037544          MOV     #$TKQSRT,$TKQIN ;;RESET THE POINTER
10719  040046  000002                  5$:     RTI                     ;;RETURN
10720
10721                                  ;;**********************************************************************
10722                                  ;*SOFTWARE SWITCH REGISTER CHANGE ROUTINE.
10723                                  ;*ROUTINE IS ENTERED  FROM THE TRAP HANDLER, AND WILL
10724                                  ;*SERVICE THE TEST FOR CHANGE IN SOFTWARE SWITCH REGISTER TRAP
10725                                  ;*CALL WHEN OPERATING IN TTY INTERRUPT MODE.
10726  040050  022737  000176  001140  $CKSWR: CMP     #SWREG,SWR      ;;IS THE SOFT-SWR SELECTED
10727  040056  001124                          BNE     15$             ;;EXIT IF NOT
10728  040060  105777  141060                  TSTB    @$TKS           ;;IS A CHAR WAITING?
10729  040064  100121                          BPL     15$             ;;IF NOT, EXIT
10730  040066  117746  141054                  MOVB    @$TKB,-(SP)     ;;YES
10731  040072  042716  177600                  BIC     #^C177,(SP)     ;;MAKE IT  7-BIT ASCII
10732  040076  021627  000007                  CMP     (SP),#7         ;;IS IT A CONTROL-G?
10733  040102  001300                          BNE     2$              ;;IF NOT, PUT IT IN THE TTY QUEUE
10734                                                                  ;;AND EXIT
10735
10736                                  ;;**********************************************************************
10737                                  ;*CONTROL IS PASSED TO THIS POINT FROM EITHER THE TTY INTERRUPT SERVICE
10738                                  ;*ROUTINE OR FROM THE SOFTWARE SWITCH REGISTER TRAP CALL, AS A RESULT OF A
10739                                  ;*CONTROL-G BEING TYPED, AND THE SOFTWARE SWITCH REGISTER BEING SELECTED.
10740  040104  123727  001134  000001  6$:     CMPB    $AUTOB,#1       ;;ARE WE RUNNING IN AUTO-MODE?
10741  040112  001674                          BEQ     2$              ;;BRANCH IF YES
10742  040114  005726                          TST     (SP)+           ;;CLEAR CONTROL-G OFF STACK
10743  040116  004737  037562                  JSR     PC,$TKINT       ;;FLUSH THE TTY INPUT QUEUE
10744  040122  005077  141016                  CLR     @$TKS           ;;DISABLE TTY KEYBOARD INTERRUPTS
10745  040126  112737  000001  001135          MOVB    #1,$INTAG       ;;SET INTERRUPT MODE INDICATOR
10746
10747  040134  104401  040633                  TYPE    ,$CNTLG         ;;ECHO THE CONTROL-G (^G)
10748  040140  104401  040640  $GTSWR: TYPE    ,$MSWR          ;;TYPE CURRENT CONTENTS
10749  040144  013746  000176                  MOV     SWREG,-(SP)     ;;SAVE SWREG FOR TYPEOUT
10750  040150  104402                          TYPOC                   ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
10751  040152  104401  040651                  TYPE    ,$MNEW          ;;PROMPT FOR NEW SWR
10752  040156  005046                  19$:    CLR     -(SP)           ;;CLEAR COUNTER
10753  040160  005046                          CLR     -(SP)           ;;THE NEW SWR
```

```
10754  040162  105777  140756          7$:     TSTB    @$TKS           ;;CHAR THERE?
10755  040166  100375                          BPL     7$              ;;IF NOT TRY AGAIN
10756
10757  040170  117746  140752                  MOVB    @$TKB,-(SP)     ;;PICK UP CHAR
10758  040174  042716  177600                  BIC     #^C177,(SP)     ;;MAKE IT 7-BIT ASCII
10759
10760  040200  021627  000003                  CMP     (SP),#3         ;;IS IT A CONTROL-C?
10761  040204  001015                          BNE     9$              ;;BRANCH IF NOT
10762  040206  104401  040621                  TYPE    ,$CNTLC         ;;YES, ECHO CONTROL-C (^C)
10763  040212  062706  000006                  ADD     #6,SP           ;;CLEAN UP STACK
10764  040216  123727  001135  000001          CMPB    $INTAG,#1       ;;REENABLE TTY KEYBOARD INTERRUPTS?
10765  040224  001003                          BNE     8$              ;;BRANCH IF NO
10766  040226  012777  000100  140710          MOV     #100,@$TKS      ;;ALLOW TTY KEYBOARD INTERRUPTS
10767  040234  000137  034450          8$:     JMP     OPERSEL         ;;CONTROL-C RESTART
10768
10769
10770  040240  021627  000025          9$:     CMP     (SP),#25        ;;IS IT A CONTROL-U?
10771  040244  001005                          BNE     10$             ;;BRANCH IF NOT
10772  040246  104401  040626                  TYPE    ,$CNTLU         ;;YES, ECHO CONTROL-U (^U)
10773  040252  062706  000006          20$:    ADD     #6,SP           ;;IGNORE PREVIOUS INPUT
10774  040256  000737                          BR      19$             ;;LET'S TRY IT AGAIN
10775
10776
10777  040260  021627  000015          10$:    CMP     (SP),#15        ;;IS IT A <CR>?
10778  040264  001022                          BNE     16$             ;;BRANCH IF NO
10779  040266  005766  000004                  TST     4(SP)           ;;YES, IS IT THE FIRST CHAR?
10780  040272  001403                          BEQ     11$             ;;BRANCH IF YES
10781  040274  016677  000002  140636          MOV     2(SP),@SWR      ;;SAVE NEW SWR
10782  040302  062706  000006          11$:    ADD     #6,SP           ;;CLEAR UP STACK
10783  040306  104401  001223          14$:    TYPE    ,$CRLF          ;;ECHO <CR> AND <LF>
10784  040312  123727  001135  000001          CMPB    $INTAG,#1       ;;RE-ENABLE TTY KBD INTERRUPTS?
10785  040320  001003                          BNE     15$             ;;BRANCH IF NOT
10786  040322  012777  000100  140614          MOV     #100,@$TKS      ;;RE-ENABLE TTY KBD INTERRUPTS
10787  040330  000002          15$:    RTI                             ;;RETURN
10788  040332  004737  037472          16$:    JSR     PC,$TYPEC       ;;ECHO CHAR
10789  040336  021627  000060                  CMP     (SP),#60        ;;CHAR < 0?
10790  040342  002420                          BLT     18$             ;;BRANCH IF YES
10791  040344  021627  000067                  CMP     (SP),#67        ;;CHAR > 7?
10792  040350  003015                          BGT     18$             ;;BRANCH IF YES
10793  040352  042726  000060                  BIC     #60,(SP)+       ;;STRIP-OFF ASCII
10794  040356  005766  000002                  TST     2(SP)           ;;IS THIS THE FIRST CHAR
10795  040362  001403                          BEQ     17$             ;;BRANCH IF YES
10796  040364  006316                          ASL     (SP)            ;;NO, SHIFT PRESENT
10797  040366  006316                          ASL     (SP)            ;;   CHAR OVER TO MAKE
10798  040370  006316                          ASL     (SP)            ;;   ROOM FOR NEW ONE.
10799  040372  005266  000002          17$:    INC     2(SP)           ;;KEEP COUNT OF CHAR
10800  040376  056616  177776                  BIS     -2(SP),(SP)     ;;SET IN NEW CHAR
10801  040402  000667                          BR      7$              ;;GET THE NEXT ONE
10802  040404  104401  001222          18$:    TYPE    ,$QUES          ;;TYPE ?<CR><LF>
10803  040410  000720                          BR      20$             ;;SIMULATE CONTROL-U
10804                                  .DSABL  LSB
10805
10806
10807                                  ;;*****************************************************************
10808                                  ;*THIS ROUTINE WILL INPUT A SINGLE CHARACTER FROM THE TTY
10809                                  ;*CALL:
```

```
10810                                   ;*      RDCHR                   ;;GET A CHARACTER FROM THE QUEUE
10811                                   ;*      RETURN HERE             ;;CHARACTER IS ON THE STACK
10812                                   ;*                              ;;WITH PARITY BIT STRIPPED OFF
10813                                   ;
10814
10815  040412  011646                  $RDCHR: MOV     (SP),-(SP)      ;;PUSH DOWN THE PC AND
10816  040414  016666  000004  000002          MOV     4(SP),2(SP)     ;;THE PS
10817  040422  005066  000004                  CLR     4(SP)           ;;GET READY FOR A CHARACTER
10818  040426  005046                          CLR     -(SP)           ;;PUT NEW PS ON STACK
10819  040430  012746  040436                  MOV     #64$,-(SP)      ;;PUT NEW PC ON  STACK
10820  040434  000002                          RTI                     ;;POP NEW PC AND PS
10821  040436                          64$:
10822  040436  005737  037542          1$:     TST     $TKCNT          ;;WAIT ON A CHARACTER
10823  040442  001775                          BEQ     1$
10824  040444  005337  037542                  DEC     $TKCNT          ;;DECREMENT THE COUNTER
10825  040450  117766  177072  000004          MOVB    @$TKQOUT,4(SP)  ;;GET ONE CHARACTER
10826  040456  005237  037546                  INC     $TKQOUT         ;;UPDATE THE POINTER
10827  040462  023727  037546  037561          CMP     $TKQOUT,#$TKQEND ;;DID IT GO OFF OF THE END?
10828  040470  001003                          BNE     2$              ;;BRANCH IF NO
10829  040472  012737  037550  037546          MOV     #$TKQSRT,$TKQOUT ;;RESET THE POINTER
10830  040500  000002                  2$:     RTI                     ;;RETURN
10831                                   ;;**********************************************************************
10832                                   ;*THIS ROUTINE WILL INPUT A STRING FROM THE TTY
10833                                   ;*CALL:
10834                                   ;*      RDLIN                   ;;INPUT A STRING FROM THE TTY
10835                                   ;*      RETURN HERE             ;;ADDRESS OF FIRST CHARACTER WILL BE ON THE STACK
10836                                   ;*                              ;;TERMINATOR WILL BE A BYTE OF ALL 0'S
10837
10838  040502  010346                  $RDLIN: MOV     R3,-(SP)        ;;SAVE R3
10839  040504  012703  040610          1$:     MOV     #$TTYIN,R3      ;;GET ADDRESS
10840  040510  022703  040621          2$:     CMP     #$TTYIN+9.,R3   ;;BUFFER FULL?
10841  040514  101405                          BLOS    4$              ;;BR IF YES
10842  040516  104410                          RDCHR                   ;;GO READ ONE CHARACTER FROM THE TTY
10843  040520  112613                          MOVB    (SP)+,(R3)      ;;GET CHARACTER
10844  040522  122713  000177          10$:    CMPB    #177,(R3)       ;;IS IT A RUBOUT
10845  040526  001003                          BNE     3$              ;;SKIP IF NOT
10846  040530  104401  001222          4$:     TYPE    ,$QUES          ;;TYPE A '?'
10847  040534  000763                          BR      1$              ;;CLEAR THE BUFFER AND LOOP
10848  040536  111337  040606          3$:     MOVB    (R3),9$         ;;ECHO THE CHARACTER
10849  040542  104401  040606                  TYPE    ,9$
10850  040546  122723  000015                  CMPB    #15,(R3)+       ;;CHECK FOR RETURN
10851  040552  001356                          BNE     2$              ;;LOOP IF NOT RETURN
10852  040554  105063  177777                  CLRB    -1(R3)          ;;CLEAR RETURN (THE 15)
10853  040560  104401  001224                  TYPE    ,$LF            ;;TYPE A LINE FEED
10854  040564  012603                          MOV     (SP)+,R3        ;;RESTORE R3
10855  040566  011646                          MOV     (SP),-(SP)      ;;ADJUST THE STACK AND PUT ADDRESS OF THE
10856  040570  016666  000004  000002          MOV     4(SP),2(SP)     ;;      FIRST ASCII CHARACTER ON IT
10857  040576  012766  040610  000004          MOV     #$TTYIN,4(SP)
10858  040604  000002                          RTI                     ;;RETURN
10859  040606     000                  9$:     .BYTE   0               ;;STORAGE FOR ASCII CHAR. TO TYPE
10860  040607     000                          .BYTE   0               ;;TERMINATOR
10861  040610  000011                  $TTYIN: .BLKB   9.              ;;RESERVE 9. BYTES FOR TTY INPUT
10862  040621     136  006503  000012 $CNTLC: .ASCIZ  /^C/<15><12>     ;;CONTROL "C"
10863  040626  052536  005015     000 $CNTLU: .ASCIZ  /^U/<15><12>     ;;CONTROL "U"
10864  040633     136  006507  000012 $CNTLG: .ASCIZ  /^G/<15><12>     ;;CONTROL "G"
10865  040640  005015  053523  020122 $MSWR:  .ASCIZ  <15><12>/SWR = /
```

```
 10866  040646  020075     000
 10867  040651     040  047040  053505  $MNEW:  .ASCIZ  /  NEW = /
 10868  040656  036440  000040
 10869                                                          ;FROM THE TTY
```

```
                                .SBTTL  READ AN OCTAL NUMBER FROM THE TTY

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

040662  011646                 $RDOCT: MOV     (SP),-(SP)      ;;PROVIDE SPACE FOR THE
040664  016666  000004  000002          MOV     4(SP),2(SP)     ;;INPUT NUMBER
040672  010046                         MOV     R0,-(SP)        ;;PUSH R0 ON STACK
040674  010146                         MOV     R1,-(SP)        ;;PUSH R1 ON STACK
040676  010246                         MOV     R2,-(SP)        ;;PUSH R2 ON STACK
040700  104411                 1$:     RDLIN                   ;;READ AN ASCIZ LINE
040702  012600                         MOV     (SP)+,R0        ;;GET ADDRESS OF 1ST CHARACTER
040704  010037  041010                 MOV     R0,5$           ;;AND SAVE IT
040710  005001                         CLR     R1              ;;CLEAR DATA WORD
040712  005002                         CLR     R2
040714  112046                 2$:     MOVB    (R0)+,-(SP)     ;;PICKUP THIS CHARACTER
040716  001420                         BEQ     3$              ;;IF ZERO GET OUT
040720  122716  000060                 CMPB    #'0,(SP)        ;;MAKE SURE THIS CHARACTER
040724  003026                         BGT     4$              ;;IS AN OCTAL DIGIT
040726  122716  000067                 CMPB    #'7,(SP)
040732  002423                         BLT     4$
040734  006301                         ASL     R1              ;;*2
040736  006102                         ROL     R2
040740  006301                         ASL     R1              ;;*4
040742  006102                         ROL     R2
040744  006301                         ASL     R1              ;;*8
040746  006102                         ROL     R2
040750  042716  177770                 BIC     #^C7,(SP)       ;;STRIP THE ASCII JUNK
040754  062601                         ADD     (SP)+,R1        ;;ADD IN THIS DIGIT
040756  000756                         BR      2$              ;;LOOP
040760  005726                 3$:     TST     (SP)+           ;;CLEAN TERMINATOR FROM STACK
040762  010166  000012                 MOV     R1,12(SP)       ;;SAVE THE RESULT
040766  010237  041020                 MOV     R2,$HIOCT
040772  012602                         MOV     (SP)+,R2        ;;POP STACK INTO R2
040774  012601                         MOV     (SP)+,R1        ;;POP STACK INTO R1
040776  012600                         MOV     (SP)+,R0        ;;POP STACK INTO R0
041000  000002                         RTI                     ;;RETURN
041002  005726                 4$:     TST     (SP)+           ;;CLEAN PARTIAL FROM STACK
041004  105010                         CLRB    (R0)            ;;SET A TERMINATOR
041006  104401                         TYPE                    ;;TYPE UP THRU THE BAD CHAR.
041010  000000                 5$:     .WORD   0
041012  104401  001222                 TYPE    ,$QUES          ;;"?" "CR" & "LF"
041016  000730                         BR      1$              ;;TRY AGAIN
041020  000000                 $HIOCT: .WORD   0               ;;HIGH ORDER BITS GO HERE
```

```
10923                                   .SBTTL  ERROR HANDLER ROUTINE
10924
10925                                   ;;*************************************************************************
10926                                   ;*THIS ROUTINE WILL INCREMENT THE ERROR FLAG AND THE ERROR COUNT,
10927                                   ;*SAVE THE ERROR ITEM NUMBER AND THE ADDRESS OF THE ERROR CALL
10928                                   ;*AND GO TO $ERRTYP ON ERROR
10929                                   ;*THE SWITCH OPTIONS PROVIDED BY THIS ROUTINE ARE:
10930                                   ;*SW15=1         HALT ON ERROR
10931                                   ;*SW13=1         INHIBIT ERROR TYPEOUTS
10932                                   ;*SW10=1         BELL ON ERROR
10933                                   ;*SW09=1         LOOP ON ERROR
10934                                   ;*CALL
10935                                   ;*      ERROR   N       ;;ERROR=EMT AND N=ERROR ITEM NUMBER
10936
10937  041022                           $ERROR:
10938  041022  104407                          CKSWR                   ;;TEST FOR CHANGE IN SOFT-SWR
10939
10940  041024                           REGSAV:
10941  041024  012737  177777  004632          MOV     #-1,@#ERFLG$    ;SET ERROR FLAG
10942  041032                           REGSA1:
10943
10944  041032  105237  001103          7$:     INCB    $ERFLG          ;;SET THE ERROR FLAG
10945  041036  001775                          BEQ     7$              ;;DON'T LET THE FLAG GO TO ZERO
10946  041040  013777  001102  140074          MOV     $TSTNM,@DISPLAY ;;DISPLAY TEST NUMBER AND ERROR FLAG
10947  041046  032777  002000  140064          BIT     #BIT10,@SWR     ;;BELL ON ERROR?
10948  041054  001402                          BEQ     1$              ;;NO - SKIP
10949  041056  104401  001216                  TYPE    ,$BELL          ;;RING BELL
10950  041062  005237  001112          1$:     INC     $ERTTL          ;;COUNT THE NUMBER OF ERRORS
10951  041066  011637  001116                  MOV     (SP),$ERRPC     ;;GET ADDRESS OF ERROR INSTRUCTION
10952  041072  162737  000002  001116          SUB     #2,$ERRPC
10953  041100  117737  140012  001114          MOVB    @$ERRPC,$ITEMB  ;;STRIP AND SAVE THE ERROR ITEM CODE
10954  041106  032777  020000  140024          BIT     #BIT13,@SWR     ;;SKIP TYPEOUT IF SET
10955  041114  001004                          BNE     20$             ;;SKIP TYPEOUTS
10956  041116  004737  041172                  JSR     PC,$ERRTYP      ;;GO TO USER ERROR ROUTINE
10957  041122  104401  001223                  TYPE    ,$CRLF
10958  041126                           20$:
10959  041126  005777  140006          2$:     TST     @SWR            ;;HALT ON ERROR
10960  041132  100002                          BPL     3$              ;;SKIP IF CONTINUE
10961  041134  000000                          HALT                    ;;HALT ON ERROR!
10962  041136  104407                          CKSWR                   ;;TEST FOR CHANGE IN SOFT-SWR
10963  041140  032777  001000  137772  3$:     BIT     #BIT09,@SWR     ;;LOOP ON ERROR SWITCH SET?
10964  041146  001402                          BEQ     4$              ;;BR IF NO
10965  041150  013716  001110                  MOV     $LPERR,(SP)     ;;FUDGE RETURN FOR LOOPING
10966  041154  005737  001214          4$:     TST     $ESCAPE         ;;CHECK FOR AN ESCAPE ADDRESS
10967  041160  001402                          BEQ     5$              ;;BR IF NONE
10968  041162  013716  001214                  MOV     $ESCAPE,(SP)    ;;FUDGE RETURN ADDRESS FOR ESCAPE
10969  041166                           5$:
10970  041166  000002                          RTI                     ;;RETURN
```

```
10971
10972                                   ;;*****************************************************************
10973
10974                                   .SBTTL  ERROR MESSAGE TYPEOUT ROUTINE
10975
10976                                   ;*THIS ROUTINE USES THE "ITEM CONTROL BYTE" ($ITEMB) TO DETERMINE WHICH
10977                                   ;*ERROR IS TO BE REPORTED.  IT THEN OBTAINS, FROM THE "ERROR TABLE" ($ERRTB),
10978                                   ;*AND REPORTS THE APPROPRIATE INFORMATION CONCERNING THE ERROR.
10979                                   ;*IT IS A COPY OF THE $ERRTYP SUBROUTINE FROM SYSMAC.
10980                                   ;*WITH ONLY MINOR CHANGES
10981                                   ;*FIRST IF SWITCH 6 IS SET AND SWITCH 8 RESET THEN
10982                                   ;*ALL REGISTER CONTENTS WILL BE TYPED BEFOR REPORTING THE ERROR
10983                                   ;*SECOND IF THE CURRENT ERROR HAS THE SAME ITEM NUMBER
10984                                   ;*AS THE PREVIOUS ERROR THEN ONLY THE DATA WILL BE TYPED
10985                                   ;*AND NOT THE ERROR MESSAGE AND HEADER.
10986
10987  041170  000000                   PRITEM: 0                       ;PREVIOUS ITEM NO. LOCATION
10988
10989  041172  017746  137742           $ERRTYP: MOV    @SWR,-(SP)      ;GET SWITCH SETTING
10990  041176  042716  177277                    BIC    #^C500,(SP)     ;KEEP ONLY SWITCH 8 AND 6
10991  041202  022726  000100                    CMP    #SW06,(SP)+     ;IS 6 SET AND 8 RESET
10992  041206  001001                            BNE    1$              ;IF NOT BRANCH
10993  041210  000402                            BR     2$              ;BRANCH IF SW 6 IS SET AND 8 RESET
10994  041212  000137  042132           1$:     JMP    @#TYPERR        ;JUMP IF SW 8 IS SET
10995                                                                   ;OR IF SW 8 IS RESET AND SW 6 IS RESET
10996  041216                           2$:
10997
10998  041216  104401  041224                    TYPE   ,65$            ;;TYPE ASCIZ STRING
10999  041222  000406                            BR     64$             ;;GET OVER THE ASCIZ
11000                                   ;;65$:   .ASCIZ  <15><12>/RHWC =  /
11001  041240                           64$:
11002  041240  013746  002254                    MOV    @#WC,-(SP)      ;GET READY TO TYPE RHWC CONTENTS
11003  041244  104402                            TYPOC
11004
11005
11006  041246  104401  041254                    TYPE   ,67$            ;;TYPE ASCIZ STRING
11007  041252  000406                            BR     66$             ;;GET OVER THE ASCIZ
11008                                   ;;67$:   .ASCIZ  <15><12>/RHBA =  /
11009  041270                           66$:
11010  041270  013746  002256                    MOV    @#BA,-(SP)      ;GET READY TO TYPE RHBA CONTENTS
11011  041274  104402                            TYPOC
11012
11013
11014  041276  104401  041304                    TYPE   ,69$            ;;TYPE ASCIZ STRING
11015  041302  000406                            BR     68$             ;;GET OVER THE ASCIZ
11016                                   ;;69$:   .ASCIZ  <15><12>/RHCS2 =  /
11017  041320                           68$:
11018  041320  013746  002260                    MOV    @#CS2,-(SP)     ;GET READY TO TYPE RHCS2 CONTENTS
11019  041324  104402                            TYPOC
11020
11021
11022  041326  104401  041334                    TYPE   ,71$            ;;TYPE ASCIZ STRING
11023  041332  000406                            BR     70$             ;;GET OVER THE ASCIZ
11024                                   ;;71$:   .ASCIZ  <15><12>/RHCS1 =  /
11025  041350                           70$:
11026  041350  013746  002262                    MOV    @#CS1,-(SP)     ;GET READY TO TYPE RHCS1 CONTENTS
```

```
11027  041354  104402                          TYPOC
11028
11029
11030  041356  104401  041364                  TYPE    ,73$            ;;TYPE ASCIZ STRING
11031  041362  000406                          BR      72$             ;;GET OVER THE ASCIZ
11032                                  ;;73$:  .ASCIZ  <15><12>/RHDS1 =  /
11033  041400                          72$:
11034  041400  013746  002304                  MOV     @#DS1,-(SP)     ;GET READY TO TYPE RHDS1 CONTENTS
11035  041404  104402                          TYPOC
11036
11037
11038  041406  104401  041414                  TYPE    ,75$            ;;TYPE ASCIZ STRING
11039  041412  000406                          BR      74$             ;;GET OVER THE ASCIZ
11040                                  ;;75$:  .ASCIZ  <15><12>/RHER1 =  /
11041  041430                          74$:
11042  041430  013746  002264                  MOV     @#ER1,-(SP)     ;GET READY TO TYPE RHER1 CONTENTS
11043  041434  104402                          TYPOC
11044
11045
11046  041436  104401  041444                  TYPE    ,77$            ;;TYPE ASCIZ STRING
11047  041442  000406                          BR      76$             ;;GET OVER THE ASCIZ
11048                                  ;;77$:  .ASCIZ  <15><12>/RHER2 =  /
11049  041460                          76$:
11050  041460  013746  002270                  MOV     @#ER2,-(SP)     ;GET READY TO TYPE RHER2 CONTENTS
11051  041464  104402                          TYPOC
11052
11053
11054  041466  104401  041474                  TYPE    ,79$            ;;TYPE ASCIZ STRING
11055  041472  000406                          BR      78$             ;;GET OVER THE ASCIZ
11056                                  ;;79$:  .ASCIZ  <15><12>/RHER3 =  /
11057  041510                          78$:
11058  041510  013746  002276                  MOV     @#ER3,-(SP)     ;GET READY TO TYPE RHER3 CONTENTS
11059  041514  104402                          TYPOC
11060
11061
11062  041516  104401  041524                  TYPE    ,81$            ;;TYPE ASCIZ STRING
11063  041522  000406                          BR      80$             ;;GET OVER THE ASCIZ
11064                                  ;;81$:  .ASCIZ  <15><12>/RHDST =  /
11065  041540                          80$:
11066  041540  013746  002266                  MOV     @#DST,-(SP)     ;GET READY TO TYPE RHDST CONTENTS
11067  041544  104402                          TYPOC
11068
11069
11070  041546  104401  041554                  TYPE    ,83$            ;;TYPE ASCIZ STRING
11071  041552  000406                          BR      82$             ;;GET OVER THE ASCIZ
11072                                  ;;83$:  .ASCIZ  <15><12>/RHCA =  /
11073  041570                          82$:
11074  041570  013746  002274                  MOV     @#CA,-(SP)      ;GET READY TO TYPE RHCA CONTENTS
11075  041574  104402                          TYPOC
11076
11077
11078  041576  104401  041604                  TYPE    ,85$            ;;TYPE ASCIZ STRING
11079  041602  000406                          BR      84$             ;;GET OVER THE ASCIZ
11080                                  ;;85$:  .ASCIZ  <15><12>/RHAS =  /
11081  041620                          84$:
11082  041620  013746  002300                  MOV     @#AS,-(SP)      ;GET READY TO TYPE RHAS CONTENTS
```

```
11083  041624  104402                          TYPOC
11084
11085
11086  041626  104401  041634                  TYPE    ,87$            ;;TYPE ASCIZ STRING
11087  041632  000406                          BR      86$             ;;GET OVER THE ASCIZ
11088                                  ;;87$:   .ASCIZ  <15><12>/RHOF =  /
11089  041650                          86$:
11090  041650  013746  002272                  MOV     @#OF,-(SP)      ;GET READY TO TYPE RHOF CONTENTS
11091  041654  104402                          TYPOC
11092
11093
11094  041656  104401  041664                  TYPE    ,89$            ;;TYPE ASCIZ STRING
11095  041662  000406                          BR      88$             ;;GET OVER THE ASCIZ
11096                                  ;;89$:   .ASCIZ  <15><12>/RHMR =  /
11097  041700                          88$:
11098  041700  013746  002302                  MOV     @#MR,-(SP)      ;GET READY TO TYPE RHMR CONTENTS
11099  041704  104402                          TYPOC
11100
11101
11102  041706  104401  041714                  TYPE    ,91$            ;;TYPE ASCIZ STRING
11103  041712  000406                          BR      90$             ;;GET OVER THE ASCIZ
11104                                  ;;91$:   .ASCIZ  <15><12>/RHLA =  /
11105  041730                          90$:
11106  041730  013746  002320                  MOV     @#LA,-(SP)      ;GET READY TO TYPE RHLA CONTENTS
11107  041734  104402                          TYPOC
11108
11109
11110  041736  104401  041744                  TYPE    ,93$            ;;TYPE ASCIZ STRING
11111  041742  000406                          BR      92$             ;;GET OVER THE ASCIZ
11112                                  ;;93$:   .ASCIZ  <15><12>/RHCC =  /
11113  041760                          92$:
11114  041760  013746  002316                  MOV     @#CC,-(SP)      ;GET READY TO TYPE RHCC CONTENTS
11115  041764  104402                          TYPOC
11116
11117
11118  041766  104401  041774                  TYPE    ,95$            ;;TYPE ASCIZ STRING
11119  041772  000406                          BR      94$             ;;GET OVER THE ASCIZ
11120                                  ;;95$:   .ASCIZ  <15><12>/RHEC1 =  /
11121  042010                          94$:
11122  042010  013746  002312                  MOV     @#EC1,-(SP)     ;GET READY TO TYPE RHEC1 CONTENTS
11123  042014  104402                          TYPOC
11124
11125
11126  042016  104401  042024                  TYPE    ,97$            ;;TYPE ASCIZ STRING
11127  042022  000406                          BR      96$             ;;GET OVER THE ASCIZ
11128                                  ;;97$:   .ASCIZ  <15><12>/RHEC2 =  /
11129  042040                          96$:
11130  042040  013746  002314                  MOV     @#EC2,-(SP)     ;GET READY TO TYPE RHEC2 CONTENTS
11131  042044  104402                          TYPOC
11132
11133
11134  042046  104401  042054                  TYPE    ,99$            ;;TYPE ASCIZ STRING
11135  042052  000406                          BR      98$             ;;GET OVER THE ASCIZ
11136                                  ;;99$:   .ASCIZ  <15><12>/RHDT =  /
11137  042070                          98$:
11138  042070  013746  002306                  MOV     @#DT,-(SP)      ;GET READY TO TYPE RHDT CONTENTS
```

```
11139  042074  104402                          TYPOC
11140
11141
11142  042076  104401  042104                  TYPE    ,101$           ;;TYPE ASCIZ STRING
11143  042102  000406                          BR      100$            ;;GET OVER THE ASCIZ
11144                                  ;;101$: .ASCIZ  <15><12>/RHSN =  /
11145  042120                          100$:
11146  042120  013746  002310                  MOV     @#SN,-(SP)      ;GET READY TO TYPE RHSN CONTENTS
11147  042124  104402                          TYPOC
11148
11149  042126  005037  041170                  CLR     @#PRITEM        ;CLEAR PREVIOUS ERROR ITEM
11150
11151  042132                          TYPERR:
11152  042132  104401  001223                  TYPE    ,$CRLF          ;"CARRIAGE RETURN" & "LINE FEED"
11153  042136  010046                          MOV     R0,-(SP)        ;SAVE R0
11154  042140  005000                          CLR     R0              ;PICKUP THE ITEM INDEX
11155  042142  153700  001114                  BISB    @#$ITEMB,R0
11156  042146  001004                          BNE     1$              ;IF ITEM NUMBER IS ZERO, JUST
11157                                                                  ;TYPE THE PC OF THE ERROR
11158  042150  013746  001116                  MOV     $ERRPC,-(SP)    ;SAVE $ERRPC FOR TYPEOUT
11159                                                                  ;ERROR ADDRESS
11160  042154  104402                          TYPOC                   ;GO TYPE--OCTAL ASCII(ALL DIGITS)
11161  042156  000454                          BR      10$             ;GET OUT
11162  042160  005300                  1$:     DEC     R0              ;ADJUST THE INDEX SO THAT IT WILL
11163  042162  006300                          ASL     R0              ;       WORK FOR THE ERROR TABLE
11164  042164  006300                          ASL     R0
11165  042166  006300                          ASL     R0
11166  042170  062700  001226                  ADD     #$ERRTB,R0      ;FORM TABLE POINTER
11167  042174  020037  041170                  CMP     R0,@#PRITEM     ;WAS PREVIOUS ERROR SAME
11168  042200  001002                          BNE     13$             ;BRANCH IF NOT
11169  042202  022020                          CMP     (R0)+,(R0)+     ;POP R0 OVER EM AND DH
11170  042204  000420                          BR      5$
11171  042206  010037  041170          13$:    MOV     R0,@#PRITEM     ;SAVE NEW ERROR ITEM
11172  042212  012037  042222                  MOV     (R0)+,2$        ;PICKUP "ERROR MESSAGE" POINTER
11173  042216  001404                          BEQ     3$              ;SKIP TYPEOUT IF NO POINTER
11174  042220  104401                          TYPE                    ;TYPE THE "ERROR MESSAGE"
11175  042222  000000                  2$:     .WORD   0               ;"ERROR MESSAGE" POINTER GOES HERE
11176  042224  104401  001223                  TYPE    ,$CRLF          ;"CARRIAGE RETURN" & "LINE FEED"
11177  042230  012037  042240          3$:     MOV     (R0)+,4$        ;PICKUP "DATA HEADER" POINTER
11178  042234  001404                          BEQ     5$              ;SKIP TYPEOUT IF 0
11179  042236  104401                          TYPE                    ;TYPE THE "DATA HEADER"
11180  042240  000000                  4$:     .WORD   0               ;"DATA HEADER" POINTER GOES HERE
11181  042242  104401  001223                  TYPE    ,$CRLF          ;"CARRIAGE RETURN" & "LINE FEED"
11182  042246  010146                  5$:     MOV     R1,-(SP)        ;SAVE R1
11183  042250  012001                          MOV     (R0)+,R1        ;PICKUP "DATA TABLE" POINTER
11184  042252  001415                          BEQ     9$              ;BR IF NO DATA TO BE TYPED
11185  042254  012000                          MOV     (R0)+,R0        ;PICKUP "DATA FORMAT" POINTER
11186  042256  105720                  6$:     TSTB    (R0)+           ;"OCTAL" OR "DECIMAL"
11187  042260  001003                          BNE     7$              ;BR IF DECIMAL
11188  042262  013146                          MOV     @(R1)+,-(SP)    ;SAVE @(R1)+ FOR TYPEOUT
11189  042264  104402                          TYPOC                   ;GO TYPE--OCTAL ASCII(ALL DIGITS)
11190  042266  000402                          BR      8$
11191  042270                          7$:
11192  042270  013146                          MOV     @(R1)+,-(SP)    ;SAVE @(R1)+ FOR TYPEOUT
11193  042272  104405                          TYPDS                   ;GO TYPE--DECIMAL ASCII WITH SIGN
11194  042274  005711                  8$:     TST     (R1)            ;IS THERE ANOTHER NUMBER?
```

```
11195  042276  001403                        BEQ     9$              ;BR IF NO
11196  042300  10440i  042314                TYPE    ,11$            ;TYPE TWO(2) SPACES
11197  042304  000764                        BR      6$              ;LOOP
11198
11199  042306  012601                9$:     MOV     (SP)+,R1        ;RESTORE R1
11200  042310  012600                10$:    MOV     (SP)+,R0        ;"CARRIAGE RETURN" & "LINE FEED"
11201  042312  000207                        RTS     PC              ;RETURN
11202  042314  020040      000       11$:    .ASCIZ  /  /            ;TWO(2) SPACES
11203          042320                        .EVEN
```

```
11204
11205                                   .SBTTL  BINARY TO OCTAL (ASCII) AND TYPE
11206
11207                                   ;;**********************************************************************
11208                                   ;*THIS ROUTINE IS USED TO CHANGE A 16-BIT BINARY NUMBER TO A 6-DIGIT
11209                                   ;*OCTAL (ASCII) NUMBER AND TYPE IT.
11210                                   ;*$TYPOS---ENTER HERE TO SETUP SUPPRESS ZEROS AND NUMBER OF DIGITS TO TYPE
11211                                   ;*CALL:
11212                                   ;*      MOV     NUM,-(SP)       ;;NUMBER TO BE TYPED
11213                                   ;*      TYPOS                   ;;CALL FOR TYPEOUT
11214                                   ;*      .BYTE   N               ;;N=1 TO 6 FOR NUMBER OF DIGITS TO TYPE
11215                                   ;*      .BYTE   M               ;;M=1 OR 0
11216                                   ;*                                      ;;1=TYPE LEADING ZEROS
11217                                   ;*                                      ;;0=SUPPRESS LEADING ZEROS
11218                                   ;*
11219                                   ;*$TYPON----ENTER HERE TO TYPE OUT WITH THE SAME PARAMETERS AS THE LAST
11220                                   ;*$TYPOS OR $TYPOC
11221                                   ;*CALL:
11222                                   ;*      MOV     NUM,-(SP)       ;;NUMBER TO BE TYPED
11223                                   ;*      TYPON                   ;;CALL FOR TYPEOUT
11224                                   ;*
11225                                   ;*$TYPOC---ENTER HERE FOR TYPEOUT OF A 16 BIT NUMBER
11226                                   ;*CALL:
11227                                   ;*      MOV     NUM,-(SP)       ;;NUMBER TO BE TYPED
11228                                   ;*      TYPOC                   ;;CALL FOR TYPEOUT
11229
11230  042320  017646  000000          $TYPOS: MOV     @(SP),-(SP)     ;;PICKUP THE MODE
11231  042324  116637  000001  042543          MOVB    1(SP),$OFILL    ;;LOAD ZERO FILL SWITCH
11232  042332  112637  042545                  MOVB    (SP)+,$OMODE+1  ;;NUMBER OF DIGITS TO TYPE
11233  042336  062716  000002                  ADD     #2,(SP)         ;;ADJUST RETURN ADDRESS
11234  042342  000406                          BR      $TYPON
11235  042344  112737  000001  042543  $TYPOC: MOVB    #1,$OFILL       ;;SET THE ZERO FILL SWITCH
11236  042352  112737  000006  042545          MOVB    #6,$OMODE+1     ;;SET FOR SIX(6) DIGITS
11237  042360  112737  000005  042542  $TYPON: MOVB    #5,$OCNT        ;;SET THE ITERATION COUNT
11238  042366  010346                          MOV     R3,-(SP)        ;;SAVE R3
11239  042370  010446                          MOV     R4,-(SP)        ;;SAVE R4
11240  042372  010546                          MOV     R5,-(SP)        ;;SAVE R5
11241  042374  113704  042545                  MOVB    $OMODE+1,R4     ;;GET THE NUMBER OF DIGITS TO TYPE
11242  042400  005404                          NEG     R4
11243  042402  062704  000006                  ADD     #6,R4           ;;SUBTRACT IT FOR MAX. ALLOWED
11244  042406  110437  042544                  MOVB    R4,$OMODE       ;;SAVE IT FOR USE
11245  042412  113704  042543                  MOVB    $OFILL,R4       ;;GET THE ZERO FILL SWITCH
11246  042416  016605  000012                  MOV     12(SP),R5       ;;PICKUP THE INPUT NUMBER
11247  042422  005003                          CLR     R3              ;;CLEAR THE OUTPUT WORD
11248  042424  006105                  1$:     ROL     R5              ;;ROTATE MSB INTO "C"
11249  042426  000404                          BR      3$              ;;GO DO MSB
11250  042430  006105                  2$:     ROL     R5              ;;FORM THIS DIGIT
11251  042432  006105                          ROL     R5
11252  042434  006105                          ROL     R5
11253  042436  010503                          MOV     R5,R3
11254  042440  006103                  3$:     ROL     R3              ;;GET LSB OF THIS DIGIT
11255  042442  105337  042544                  DECB    $OMODE          ;;TYPE THIS DIGIT?
11256  042446  100016                          BPL     7$              ;;BR IF NO
11257  042450  042703  177770                  BIC     #177770,R3      ;;GET RID OF JUNK
11258  042454  001002                          BNE     4$              ;;TEST FOR 0
11259  042456  005704                          TST     R4              ;;SUPPRESS THIS 0?
```

```
11260  042460  001403                        BEQ     5$              ;;BR IF YES
11261  042462  005204                4$:     INC     R4              ;;DON'T SUPPRESS ANYMORE 0'S
11262  042464  052703  000060                BIS     #'0,R3          ;;MAKE THIS DIGIT ASCII
11263  042470  052703  000040        5$:     BIS     #' ,R3          ;;MAKE ASCII IF NOT ALREADY
11264  042474  110337  042540                MOVB    R3,8$           ;;SAVE FOR TYPING
11265  042500  104401  042540                TYPE    ,8$             ;;GO TYPE THIS DIGIT
11266  042504  105337  042542        7$:     DECB    $OCNT           ;;COUNT BY 1
11267  042510  003347                        BGT     2$              ;;BR IF MORE TO DO
11268  042512  002402                        BLT     6$              ;;BR IF DONE
11269  042514  005204                        INC     R4              ;;INSURE LAST DIGIT ISN'T A BLANK
11270  042516  000744                        BR      2$              ;;GO DO THE LAST DIGIT
11271  042520  012605                6$:     MOV     (SP)+,R5        ;;RESTORE R5
11272  042522  012604                        MOV     (SP)+,R4        ;;RESTORE R4
11273  042524  012603                        MOV     (SP)+,R3        ;;RESTORE R3
11274  042526  016666  000002  000004        MOV     2(SP),4(SP)     ;;SET THE STACK FOR RETURNING
11275  042534  012616                        MOV     (SP)+,(SP)
11276  042536  000002                        RTI                     ;;RETURN
11277  042540     000                8$:     .BYTE   0               ;;STORAGE FOR ASCII DIGIT
11278  042541     000                        .BYTE   0               ;;TERMINATOR FOR TYPE ROUTINE
11279  042542     000                $OCNT:  .BYTE   0               ;;OCTAL DIGIT COUNTER
11280  042543     000                $OFILL: .BYTE   0               ;;ZERO FILL SWITCH
11281  042544  000000                $OMODE: .WORD   0               ;;NUMBER OF DIGITS TO TYPE
```

```
11282                                   .SBTTL  TRAP DECODER
11283
11284                                   ;;**********************************************************************
11285                                   ;*THIS ROUTINE WILL PICKUP THE LOWER BYTE OF THE "TRAP" INSTRUCTION
11286                                   ;*AND USE IT TO INDEX THROUGH THE TRAP TABLE FOR THE STARTING ADDRESS
11287                                   ;*OF THE DESIRED ROUTINE. THEN USING THE ADDRESS OBTAINED IT WILL
11288                                   ;*GO TO THAT ROUTINE.
11289
11290  042546  010046                   $TRAP:  MOV     R0,-(SP)        ;;SAVE R0
11291  042550  016600  000002                   MOV     2(SP),R0        ;;GET TRAP ADDRESS
11292  042554  005740                           TST     -(R0)           ;;BACKUP BY 2
11293  042556  111000                           MOVB    (R0),R0         ;;GET RIGHT BYTE OF TRAP
11294  042560  006300                           ASL     R0              ;;POSITION FOR INDEXING
11295  042562  016000  042602                   MOV     $TRPAD(R0),R0   ;;INDEX TO TABLE
11296  042566  000200                           RTS     R0              ;;GO TO ROUTINE
11297
11298
11299                                   ;;THIS IS USE TO HANDLE THE "GETPRI" MACRO
11300
11301  042570  011646                   $TRAP2: MOV     (SP),-(SP)      ;;MOVE THE PC DOWN
11302  042572  016666  000004  000002           MOV     4(SP),2(SP)     ;;MOVE THE PSW DOWN
11303  042600  000002                           RTI                     ;;RESTORE THE PSW
11304
11305                                   .SBTTL  TRAP TABLE
11306
11307                                   ;*THIS TABLE CONTAINS THE STARTING ADDRESSES OF THE ROUTINES CALLED
11308                                   ;*BY THE "TRAP" INSTRUCTION.
11309
11310                                   ;       ROUTINE
11311                                   ;       -------
11312  042602  042570                   $TRPAD: .WORD   $TRAP2
11313  042604  037322                           $TYPE   ;;CALL=TYPE     TRAP+1(104401)  TTY TYPEOUT ROUTINE
11314  042606  042344                           $TYPOC  ;;CALL=TYPOC    TRAP+2(104402)  TYPE OCTAL NUMBER (WITH LEADING ZEROS)
11315  042610  042320                           $TYPOS  ;;CALL=TYPOS    TRAP+3(104403)  TYPE OCTAL NUMBER (NO LEADING ZEROS)
11316  042612  042360                           $TYPON  ;;CALL=TYPON    TRAP+4(104404)  TYPE OCTAL NUMBER (AS PER LAST CALL)
11317  042614  037076                           $TYPDS  ;;CALL=TYPDS    TRAP+5(104405)  TYPE DECIMAL NUMBER (WITH SIGN)
11318
11319  042616  040140                           $GTSWR  ;;CALL=GTSWR    TRAP+6(104406)  GET SOFT-SWR SETTING
11320
11321  042620  040050                           $CKSWR  ;;CALL=CKSWR    TRAP+7(104407)  TEST FOR CHANGE IN SOFT-SWR
11322  042622  040412                           $RDCHR  ;;CALL=RDCHR    TRAP+10(104410) TTY TYPEIN CHARACTER ROUTINE
11323  042624  040502                           $RDLIN  ;;CALL=RDLIN    TRAP+11(104411) TTY TYPEIN STRING ROUTINE
11324  042626  040662                           $RDOCT  ;;CALL=RDOCT    TRAP+12(104412) READ AN OCTAL NUMBER FROM TTY
11325  042630  034016                           WAIT.T  ;;CALL=WAT      TRAP+13(104413) DONT ADD ABOVE THIS TRAP
```

```
11326                                   .SBTTL  POWER DOWN AND UP ROUTINES
11327
11328                                   ;;**********************************************************************
11329                                   ;POWER DOWN ROUTINE
11330  042632  012737  042776  000024  $PWRDN: MOV     #$ILLUP,@#PWRVEC ;;SET FOR FAST UP
11331  042640  012737  000340  000026          MOV     #340,@#PWRVEC+2 ;;PRIO:7
11332  042646  010046                         MOV     R0,-(SP)        ;;PUSH R0 ON STACK
11333  042650  010146                         MOV     R1,-(SP)        ;;PUSH R1 ON STACK
11334  042652  010246                         MOV     R2,-(SP)        ;;PUSH R2 ON STACK
11335  042654  010346                         MOV     R3,-(SP)        ;;PUSH R3 ON STACK
11336  042656  010446                         MOV     R4,-(SP)        ;;PUSH R4 ON STACK
11337  042660  010546                         MOV     R5,-(SP)        ;;PUSH R5 ON STACK
11338  042662  017746  136252                 MOV     @SWR,-(SP)      ;;PUSH @SWR ON STACK
11339  042666  010637  043002                 MOV     SP,$SAVR6       ;;SAVE SP
11340  042672  012737  042704  000024         MOV     #$PWRUP,@#PWRVEC ;;SET UP VECTOR
11341  042700  000000                         HALT
11342  042702  000776                         BR      .-2             ;;HANG UP
11343
11344                                   ;;**********************************************************************
11345                                   ;POWER UP ROUTINE
11346  042704  012737  042776  000024  $PWRUP: MOV     #$ILLUP,@#PWRVEC ;;SET FOR FAST DOWN
11347  042712  013706  043002                 MOV     $SAVR6,SP       ;;GET SP
11348  042716  005037  043002                 CLR     $SAVR6          ;;WAIT LOOP FOR THE TTY
11349  042722  005237  043002          1$:    INC     $SAVR6          ;;WAIT FOR THE INC
11350  042726  001375                         BNE     1$              ;;OF  WORD
11351  042730  012677  136204                 MOV     (SP)+,@SWR      ;;POP STACK INTO @SWR
11352  042734  012605                         MOV     (SP)+,R5        ;;POP STACK INTO R5
11353  042736  012604                         MOV     (SP)+,R4        ;;POP STACK INTO R4
11354  042740  012603                         MOV     (SP)+,R3        ;;POP STACK INTO R3
11355  042742  012602                         MOV     (SP)+,R2        ;;POP STACK INTO R2
11356  042744  012601                         MOV     (SP)+,R1        ;;POP STACK INTO R1
11357  042746  012600                         MOV     (SP)+,R0        ;;POP STACK INTO R0
11358  042750  012737  042632  000024         MOV     #$PWRDN,@#PWRVEC ;;SET UP THE POWER DOWN VECTOR
11359  042756  012737  000340  000026         MOV     #340,@#PWRVEC+2 ;;PRIO:7
11360  042764  104401                         TYPE                    ;REPORT THE POWER FAILURE
11361  042766  043004          $PWRMG: .WORD   $POWER          ;;POWER FAIL MESSAGE POINTER
11362  042770  012716                  MOV     (PC)+,(SP)      ;;RESTART AT BEGIN
11363  042772  004712          $PWRAD: .WORD   BEGIN           ;;RESTART ADDRESS
11364  042774  000002                  RTI
11365  042776  000000          $ILLUP: HALT                    ;;THE POWER UP SEQUENCE WAS STARTED
11366  043000  000776                  BR      .-2             ;; BEFORE THE POWER DOWN WAS COMPLETE
11367  043002  000000          $SAVR6: 0                       ;;PUT THE SP HERE
11368  043004  005015  047520  042527  $POWER: .ASCIZ  <15><12>''POWER''
11369  043012  000122
11370                                           .EVEN
11371
```

```
11372
11373                                   ;*****************************************************************************
11374                                   ;
11375                                   ;ERROR AND MESSAGE TABLE CONDIMENTS
11376                                   ;
11377                                   ;*****************************************************************************
11378
11379
11380
11381
11382  043014  050122  032060  042040  EM1:    .ASCIZ  /RP04 DID NOT INTERRUPT/
11383  043022  042111  047040  052117
11384  043030  044440  052116  051105
11385  043036  052522  052120     000
11386  043043     111  052116  051105  EM2:    .ASCIZ  /INTERRUPT ENABLE BIT DOWN BUT EXPECTED BIT DID NOT SET/
11387  043050  052522  052120  042440
11388  043056  040516  046102  020105
11389  043064  044502  020124  047504
11390  043072  047127  041040  052125
11391  043100  042440  050130  041505
11392  043106  042524  020104  044502
11393  043114  020124  044504  020104
11394  043122  047516  020124  042523
11395  043130  000124
11396  043132  050122  032060  042040  EM3:    .ASCIZ  /RP04 DID NOT INTERRUPT WHEN EXPECTED BIT DID SET/
11397  043140  042111  047040  052117
11398  043146  044440  052116  051105
11399  043154  052522  052120  053440
11400  043162  042510  020116  054105
11401  043170  042520  052103  042105
11402  043176  041040  052111  042040
11403  043204  042111  051440  052105
11404  043212     000
11405  043213     105  050130  041505  EM4:    .ASCIZ  /EXPECTED BIT DID SET BUT TIME IS IN ERROR - TIME IN 10 MICROSEC. DECIMA
11406  043220  042524  020104  044502
11407  043226  020124  044504  020104
11408  043234  042523  020124  052502
11409  043242  020124  044524  042515
11410  043250  044440  020123  047111
11411  043256  042440  051122  051117
11412  043264  026440  052040  046511
11413  043272  020105  047111  030440
11414  043300  020060  044515  051103
11415  043306  051517  041505  020056
11416  043314  042504  044503  040515
11417  043322  000114
11418  043324  044122  051501  042040  EM5:    .ASCIZ  /RHAS DOES NOT CLEAR BY MOVING IN ALL ONES/
11419  043332  042517  020123  047516
11420  043340  020124  046103  040505
11421  043346  020122  054502  046440
11422  043354  053117  047111  020107
11423  043362  047111  040440  046114
11424  043370  047440  042516  000123
11425  043376  047514  042101  047111  EM6:    .ASCIZ  /LOADING RHER1 FOR ALL UNITS DID NOT SET ANY RHAS BITS/
11426  043404  020107  044122  051105
11427  043412  020061  047506  020122
```

```
11428  043420  046101  020114  047125
11429  043426  052111  020123  044504
11430  043434  020104  047516  020124
11431  043442  042523  020124  047101
11432  043450  020131  044122  051501
11433  043456  041040  052111  000123
11434  043464  047516  020116  054105  EM7:    .ASCIZ  /NON EXISTENT REGISTER, PROGRAM ABORTED./
11435  043472  051511  042524  052116
11436  043500  051040  043505  051511
11437  043506  042524  026122  050040
11438  043514  047522  051107  046501
11439  043522  040440  047502  052122
11440  043530  042105  000056
11441  043534  052123  050117  042520  EM10:   .ASCIZ  /STOPPED DRIVE HAS MOL BIT IN RHDS1 SET/
11442  043542  020104  051104  053111
11443  043550  020105  040510  020123
11444  043556  047515  020114  044502
11445  043564  020124  047111  051040
11446  043572  042110  030523  051440
11447  043600  052105     000
11448
11449  043603     127  052111  020110  EM11:   .ASCIZ  /WITH SPINDLE POWERED DOWN RHCS2 SHOULD ONLY HAVE UNIT NO: AND IR SET/
11450  043610  050123  047111  046104
11451  043616  020105  047520  042527
11452  043624  042522  020104  047504
11453  043632  047127  051040  041510
11454  043640  031123  051440  047510
11455  043646  046125  020104  047117
11456  043654  054514  044040  053101
11457  043662  020105  047125  052111
11458  043670  047040  035117  040440
11459  043676  042116  044440  020122
11460  043704  042523  000124
11461  043710  043101  042524  020122  EM12:   .ASCIZ  /AFTER SPINDLE POWERED UP, NO PACK ACKN. RHDS1 SHOULD HAVE MOL=1, VV=0/
11462  043716  050123  047111  046104
11463  043724  020105  047520  042527
11464  043732  042522  020104  050125
11465  043740  020054  047516  050040
11466  043746  041501  020113  041501
11467  043754  047113  020056  044122
11468  043762  051504  020061  044123
11469  043770  052517  042114  044040
11470  043776  053101  020105  047515
11471  044004  036514  026061  053040
11472  044012  036526  000060
11473  044016  044527  044124  051440  EM13:   .ASCIZ  /WITH SPINDLE POWERED, NO INTIALIZE, RHCS1 SHOULD HAVE GO=0, DVA=1, RDY=
11474  044024  044520  042116  042514
11475  044032  050040  053517  051105
11476  044040  042105  020054  047516
11477  044046  044440  052116  040511
11478  044054  044514  042532  020054
11479  044062  044122  051503  020061
11480  044070  044123  052517  042114
11481  044076  044040  053101  020105
11482  044104  047507  030075  020054
11483  044112  053104  036501  026061
```

```
11484  044120  051040  054504  030475
11485  044126  020054  042511  030075
11486  044134     000
11487  044135     101  052106  051105  EM14:   .ASCIZ  /AFTER SPINDLE POWERED UP RHCC SHOULD BE=0/
11488  044142  051440  044520  042116
11489  044150  042514  050040  053517
11490  044156  051105  042105  052440
11491  044164  020120  044122  041503
11492  044172  051440  047510  046125
11493  044200  020104  042502  030075
11494  044206     000
11495  044207     120  041501  020113  EM15:   .ASCII  /PACK ACKNOWLEDGE COMMAND CAUSED AN ERROR/<15><12>
11496  044214  041501  047113  053517
11497  044222  042514  043504  020105
11498  044230  047503  046515  047101
11499  044236  020104  040503  051525
11500  044244  042105  040440  020116
11501  044252  051105  047522  006522
11502  044260     012
11503  044261     107  047517  020104          .ASCIZ  /GOOD DATA IS BEFORE COMMAND, REC DATA IS AFTER COMMAND/
11504  044266  040504  040524  044440
11505  044274  020123  042502  047506
11506  044302  042522  041440  046517
11507  044310  040515  042116  020054
11508  044316  042522  020103  040504
11509  044324  040524  044440  020123
11510  044332  043101  042524  020122
11511  044340  047503  046515  047101
11512  044346  000104
11513  044350  047516  047455  020120  EM16:   .ASCII  /NO-OP COMMAND CAUSED AN ERROR/<15><12>
11514  044356  047503  046515  047101
11515  044364  020104  040503  051525
11516  044372  042105  040440  020116
11517  044400  051105  047522  006522
11518  044406     012
11519  044407     107  047517  020104          .ASCIZ  /GOOD DATA IS BEFORE COMMAND, REC DATA IS AFTER COMMAND/
11520  044414  040504  040524  044440
11521  044422  020123  042502  047506
11522  044430  042522  041440  046517
11523  044436  040515  042116  020054
11524  044444  042522  020103  040504
11525  044452  040524  044440  020123
11526  044460  043101  042524  020122
11527  044466  047503  046515  047101
11528  044474  000104
11529  044476  051104  053111  020105  EM17:   .ASCII  /DRIVE CLEAR COMMAND CAUSED AN ERROR/<15><12>
11530  044504  046103  040505  020122
11531  044512  047503  046515  047101
11532  044520  020104  040503  051525
11533  044526  042105  040440  020116
11534  044534  051105  047522  006522
11535  044542     012
11536  044543     107  047517  020104          .ASCIZ  /GOOD DATA GIVES SHOULD BE, REC DATA GIVES AFTER COMMAND/
11537  044550  040504  040524  043440
11538  044556  053111  051505  051440
11539  044564  047510  046125  020104
```

```
11540  044572  042502  020054  042522
11541  044600  020103  040504  040524
11542  044606  043440  053111  051505
11543  044614  040440  052106  051105
11544  044622  041440  046517  040515
11545  044630  042116     000
11546  044633     122  040505  026504  EM20:   .ASCII  /READ-IN COMMAND CAUSED AN ERROR/<15><12>
11547  044640  047111  041440  046517
11548  044646  040515  042116  041440
11549  044654  052501  042523  020104
11550  044662  047101  042440  051122
11551  044670  051117  005015
11552  044674  047507  042117  042040          .ASCIZ  /GOOD DATA GIVES SHOULD BE, REC DATA GIVES REG. CONTENTS AFTER COMMAND/
11553  044702  052101  020101  044507
11554  044710  042526  020123  044123
11555  044716  052517  042114  041040
11556  044724  026105  051040  041505
11557  044732  042040  052101  020101
11558  044740  044507  042526  020123
11559  044746  042522  027107  041440
11560  044754  047117  042524  052116
11561  044762  020123  043101  042524
11562  044770  020122  047503  046515
11563  044776  047101  000104
11564
11565  045002  044122  051503  020061  EM21:   .ASCIZ  /RHCS1 CONTENTS DURING COMMAND WAS IN ERROR/
11566  045010  047503  052116  047105
11567  045016  051524  042040  051125
11568  045024  047111  020107  047503
11569  045032  046515  047101  020104
11570  045040  040527  020123  047111
11571  045046  042440  051122  051117
11572  045054     000
11573  045055     122  042110  030523  EM22:   .ASCIZ  /RHDS1 CONTENTS DURING COMMAND WAS IN ERROR/
11574  045062  041440  047117  042524
11575  045070  052116  020123  052504
11576  045076  044522  043516  041440
11577  045104  046517  040515  042116
11578  045112  053440  051501  044440
11579  045120  020116  051105  047522
11580  045126  000122
11581  045130  047125  047514  042101  EM23:   .ASCII  /UNLOAD COMMAND CAUSED AN ERROR/<15><12>
11582  045136  041440  046517  040515
11583  045144  042116  041440  052501
11584  045152  042523  020104  047101
11585  045160  042440  051122  051117
11586  045166  005015
11587  045170  047507  042117  042040          .ASCIZ  /GOOD DATA GIVES SHOULD BE, REC DATA GIVES REGISTER CONT. AFTER COMMAND/
11588  045176  052101  020101  044507
11589  045204  042526  020123  044123
11590  045212  052517  042114  041040
11591  045220  026105  051040  041505
11592  045226  042040  052101  020101
11593  045234  044507  042526  020123
11594  045242  042522  044507  052123
11595  045250  051105  041440  047117
```

```
11596  045256  027124  040440  052106
11597  045264  051105  041440  046517
11598  045272  040515  042116     000
11599  045277     117  043106  042523  EM24:   .ASCII  /OFFSET COMMAND CAUSED AN ERROR/<15><12>
11600  045304  020124  047503  046515
11601  045312  047101  020104  040503
11602  045320  051525  042105  040440
11603  045326  020116  051105  047522
11604  045334  006522     012
11605  045337     107  047517  020104          .ASCIZ  /GOOD DATA GIVES SHOULD BE, REC DATA GIVES REG. CONT. AFTER COMMAND/
11606  045344  040504  040524  043440
11607  045352  053111  051505  051440
11608  045360  047510  046125  020104
11609  045366  042502  020054  042522
11610  045374  020103  040504  040524
11611  045402  043440  053111  051505
11612  045410  051040  043505  020056
11613  045416  047503  052116  020056
11614  045424  043101  042524  020122
11615  045432  047503  046515  047101
11616  045440  000104
11617  045442  042522  052524  047122  EM25:   .ASCII  /RETURN TO CENTER LINE COMMAND CAUSED AN ERROR/<15><12>
11618  045450  052040  020117  042503
11619  045456  052116  051105  046040
11620  045464  047111  020105  047503
11621  045472  046515  047101  020104
11622  045500  040503  051525  042105
11623  045506  040440  020116  051105
11624  045514  047522  006522     012
11625  045521     107  047517  020104          .ASCIZ  /GOOD DATA GIVES SHOULD BE, REC DATA GIVES REG. CONT. AFTER COMMAND/
11626  045526  040504  040524  043440
11627  045534  053111  051505  051440
11628  045542  047510  046125  020104
11629  045550  042502  020054  042522
11630  045556  020103  040504  040524
11631  045564  043440  053111  051505
11632  045572  051040  043505  020056
11633  045600  047503  052116  020056
11634  045606  043101  042524  020122
11635  045614  047503  046515  047101
11636  045622  000104
11637  045624  030065  020060  043117  EM26:   .ASCIZ  /500 OFFSET COMMANDS ONE AFTER THE OTHER CAUSED AN ERROR/
11638  045632  051506  052105  041440
11639  045640  046517  040515  042116
11640  045646  020123  047117  020105
11641  045654  043101  042524  020122
11642  045662  044124  020105  052117
11643  045670  042510  020122  040503
11644  045676  051525  042105  040440
11645  045704  020116  051105  047522
11646  045712  000122
11647  045714  051127  052111  020105  EM27:   .ASCII  /WRITE HEADER AND DATA CAUSED IMPROPER REGISTER CHANGE/<15><12>
11648  045722  042510  042101  051105
11649  045730  040440  042116  042040
11650  045736  052101  020101  040503
11651  045744  051525  042105  044440
```

```
11652  045752  050115  047522  042520
11653  045760  020122  042522  044507
11654  045766  052123  051105  041440
11655  045774  040510  043516  006505
11656  046002     012
11657  046003     107  047517  020104          .ASCII  /GOOD DATA GIVES WHAT SHOULD BE THERE/<15><12>
11658  046010  040504  040524  043440
11659  046016  053111  051505  053440
11660  046024  040510  020124  044123
11661  046032  052517  042114  041040
11662  046040  020105  044124  051105
11663  046046  006505     012
11664  046051     122  041505  044505          .ASCIZ  /RECEIVED DATA GIVES WHAT WAS THERE AFTER COMMAND/
11665  046056  042526  020104  040504
11666  046064  040524  043440  053111
11667  046072  051505  053440  040510
11668  046100  020124  040527  020123
11669  046106  044124  051105  020105
11670  046114  043101  042524  020122
11671  046122  047503  046515  047101
11672  046130  000104
11673  046132  051127  052111  020105  EM30:   .ASCIZ  /WRITE HEADER AND DATA CHANGED WRITE FROM BUFFER/
11674  046140  042510  042101  051105
11675  046146  040440  042116  042040
11676  046154  052101  020101  044103
11677  046162  047101  042507  020104
11678  046170  051127  052111  020105
11679  046176  051106  046517  041040
11680  046204  043125  042506  000122
11681
11682  046212  042522  042101  044040  EM31:   .ASCII  /READ HEADER AND DATA CAUSED IMPROPER REGISTER CHANGE/<15><12>
11683  046220  040505  042504  020122
11684  046226  047101  020104  040504
11685  046234  040524  041440  052501
11686  046242  042523  020104  046511
11687  046250  051120  050117  051105
11688  046256  051040  043505  051511
11689  046264  042524  020122  044103
11690  046272  047101  042507  005015
11691  046300  047507  042117  042040          .ASCII  /GOOD DATA GIVES WHAT SHOULD BE THERE/<15><12>
11692  046306  052101  020101  044507
11693  046314  042526  020123  044127
11694  046322  052101  051440  047510
11695  046330  046125  020104  042502
11696  046336  052040  042510  042522
11697  046344  005015
11698  046346  042522  042503  053111          .ASCIZ  /RECEIVED DATA GIVES WHAT WAS THERE AFTER COMMAND/
11699  046354  042105  042040  052101
11700  046362  020101  044507  042526
11701  046370  020123  044127  052101
11702  046376  053440  051501  052040
11703  046404  042510  042522  040440
11704  046412  052106  051105  041440
11705  046420  046517  040515  042116
11706  046426     000
11707  046427     127  044522  042524  EM32:   .ASCIZ  /WRITE HEADER DATA FOLLOWED BY READ HEADER AND DATA CAUSED DATA ERROR/
```

```
 11708  046434  044040  040505  042504
 11709  046442  020122  040504  040524
 11710  046450  043040  046117  047514
 11711  046456  042527  020104  054502
 11712  046464  051040  040505  020104
 11713  046472  042510  042101  051105
 11714  046500  040440  042116  042040
 11715  046506  052101  020101  040503
 11716  046514  051525  042105  042040
 11717  046522  052101  020101  051105
 11718  046530  047522  000122
 11719  046534  042522  042101  042040  EM33:   .ASCII  /READ DATA CAUSED IMPROPER REGISTER CHANGE/<15><12>
 11720  046542  052101  020101  040503
 11721  046550  051525  042105  044440
 11722  046556  050115  047522  042520
 11723  046564  020122  042522  044507
 11724  046572  052123  051105  041440
 11725  046600  040510  043516  006505
 11726  046606     012
 11727  046607     107  047517  020104          .ASCII  /GOOD DATA GIVES WHAT SHOULD BE THERE/<15><12>
 11728  046614  040504  040524  043440
 11729  046622  053111  051505  053440
 11730  046630  040510  020124  044123
 11731  046636  052517  042114  041040
 11732  046644  020105  044124  051105
 11733  046652  006505     012
 11734  046655     122  041505  044505          .ASCIZ  /RECEIVED DATA GIVES WHAT WAS THERE AFTER COMMAND/
 11735  046662  042526  020104  040504
 11736  046670  040524  043440  053111
 11737  046676  051505  053440  040510
 11738  046704  020124  040527  020123
 11739  046712  044124  051105  020105
 11740  046720  043101  042524  020122
 11741  046726  047503  046515  047101
 11742  046734  000104
 11743  046736  042522  042101  042040  EM34:   .ASCIZ  /READ DATA INCORRECT/
 11744  046744  052101  020101  047111
 11745  046752  047503  051122  041505
 11746  046760  000124
 11747  046762  051127  052111  020105  EM35:   .ASCII  /WRITE DATA COMMAND CAUSED IMPROPER REGISTER CHANGE/<15><12>
 11748  046770  040504  040524  041440
 11749  046776  046517  040515  042116
 11750  047004  041440  052501  042523
 11751  047012  020104  046511  051120
 11752  047020  050117  051105  051040
 11753  047026  043505  051511  042524
 11754  047034  020122  044103  047101
 11755  047042  042507  005015
 11756  047046  047507  042117  042040          .ASCII  /GOOD DATA GIVES WHAT SHOULD BE THERE/<15><12>
 11757  047054  052101  020101  044507
 11758  047062  042526  020123  044127
 11759  047070  052101  051440  047510
 11760  047076  046125  020104  042502
 11761  047104  052040  042510  042522
 11762  047112  005015
 11763  047114  042522  042503  053111          .ASCIZ  /RECEIVED DATA GIVES REGISTER CONTENTS AFTER COMMAND/
```

```
11764  047122  042105  042040  052101
11765  047130  020101  044507  042526
11766  047136  020123  042522  044507
11767  047144  052123  051105  041440
11768  047152  047117  042524  052116
11769  047160  020123  043101  042524
11770  047166  020122  047503  046515
11771  047174  047101  000104
11772  047200  051127  052111  020105  EM36:   .ASCIZ  /WRITE DATA COMMAND CHANGED WRITE FROM BUFFER/
11773  047206  040504  040524  041440
11774  047214  046517  040515  042116
11775  047222  041440  040510  043516
11776  047230  042105  053440  044522
11777  047236  042524  043040  047522
11778  047244  020115  052502  043106
11779  047252  051105     000
11780  047255     123  042505  020113  EM37:   .ASCII  /SEEK COMMAND CAUSED IMPROPER REGISTER CHANGE/<15><12>
11781  047262  047503  046515  047101
11782  047270  020104  040503  051525
11783  047276  042105  044440  050115
11784  047304  047522  042520  020122
11785  047312  042522  044507  052123
11786  047320  051105  041440  040510
11787  047326  043516  006505     012
11788  047333     107  047517  020104          .ASCII  /GOOD DATA GIVES WHAT SHOULD BE THERE/<15><12>
11789  047340  040504  040524  043440
11790  047346  053111  051505  053440
11791  047354  040510  020124  044123
11792  047362  052517  042114  041040
11793  047370  020105  044124  051105
11794  047376  006505     012
11795  047401     122  041505  044505          .ASCIZ  /RECEIVED DATA GIVES REGISTER CONTENTS AFTER SEEK COMMAND/
11796  047406  042526  020104  040504
11797  047414  040524  043440  053111
11798  047422  051505  051040  043505
11799  047430  051511  042524  020122
11800  047436  047503  052116  047105
11801  047444  051524  040440  052106
11802  047452  051105  051440  042505
11803  047460  020113  047503  046515
11804  047466  047101  000104
11805  047472  051127  052111  020105  EM40:   .ASCII  /WRITE CHECK CAUSED IMPROPER REGISTER CHANGE/<15><12>
11806  047500  044103  041505  020113
11807  047506  040503  051525  042105
11808  047514  044440  050115  047522
11809  047522  042520  020122  042522
11810  047530  044507  052123  051105
11811  047536  041440  040510  043516
11812  047544  006505     012
11813  047547     107  047517  020104          .ASCII  /GOOD DATA GIVES WHAT SHOULD BE THERE/<15><12>
11814  047554  040504  040524  043440
11815  047562  053111  051505  053440
11816  047570  040510  020124  044123
11817  047576  052517  042114  041040
11818  047604  020105  044124  051105
11819  047612  006505     012
```

```
11820  047615    122  041505  044505          .ASCIZ  /RECEIVED DATA GIVES REGISTER CONTENTS AFTER COMMAND/
11821  047622  042526  020104  040504
11822  047630  040524  043440  053111
11823  047636  051505  051040  043505
11824  047644  051511  042524  020122
11825  047652  047503  052116  047105
11826  047660  051524  040440  052106
11827  047666  051105  041440  046517
11828  047674  040515  042116     000
11829
11830  047701    114  041517  044513  EM41:   .ASCII  /LOCKING OUT WRITE BY WRITE LOCK BUTTON CAUSED IMPROPER REGISTER CHANGE/
11831  047706  043516  047440  052125
11832  047714  053440  044522  042524
11833  047722  041040  020131  051127
11834  047730  052111  020105  047514
11835  047736  045503  041040  052125
11836  047744  047524  020116  040503
11837  047752  051525  042105  044440
11838  047760  050115  047522  042520
11839  047766  020122  042522  044507
11840  047774  052123  051105  041440
11841  050002  040510  043516  006505
11842  050010     012
11843  050011    107  047517  020104          .ASCII  /GOOD DATA GIVES WHAT SHOULD BE THERE/<15><12>
11844  050016  040504  040524  043440
11845  050024  053111  051505  053440
11846  050032  040510  020124  044123
11847  050040  052517  042114  041040
11848  050046  020105  044124  051105
11849  050054  006505     012
11850  050057    122  041505  044505          .ASCIZ  /RECEIVED DATA GIVES REGISTER CONTENTS AFTER WRITES WERE LOCKED OUT/
11851  050064  042526  020104  040504
11852  050072  040524  043440  053111
11853  050100  051505  051040  043505
11854  050106  051511  042524  020122
11855  050114  047503  052116  047105
11856  050122  051524  040440  052106
11857  050130  051105  053440  044522
11858  050136  042524  020123  042527
11859  050144  042522  046040  041517
11860  050152  042513  020104  052517
11861  050160  000124
11862  050162  052101  042524  050115  EM42:   .ASCII  /ATTEMPTING TO WRITE WITH WRITES LOCKED OUT CAUSED IMPROPER REGISTER CHA
11863  050170  044524  043516  052040
11864  050176  020117  051127  052111
11865  050204  020105  044527  044124
11866  050212  053440  044522  042524
11867  050220  020123  047514  045503
11868  050226  042105  047440  052125
11869  050234  041440  052501  042523
11870  050242  020104  046511  051120
11871  050250  050117  051105  051040
11872  050256  043505  051511  042524
11873  050264  020122  044103  047101
11874  050272  042507  005015
11875  050276  047507  042117  042040          .ASCII  /GOOD DATA GIVES WHAT SHOULD BE THERE/<15><12>
```

```
 11876  050304  052101  020101  044507
 11877  050312  042526  020123  044127
 11878  050320  052101  051440  047510
 11879  050326  046125  020104  042502
 11880  050334  052040  042510  042522
 11881  050342  005015
 11882  050344  042522  042503  053111          .ASCIZ  /RECEIVED DATA GIVES REGISTER CONTENTS AFTER ATTEMPTED WRITE/
 11883  050352  042105  042040  052101
 11884  050360  020101  044507  042526
 11885  050366  020123  042522  044507
 11886  050374  052123  051105  041440
 11887  050402  047117  042524  052116
 11888  050410  020123  043101  042524
 11889  050416  020122  052101  042524
 11890  050424  050115  042524  020104
 11891  050432  051127  052111  000105
 11892  050440  051127  052111  047111  EM43:   .ASCII  /WRITING WITH WRITES LOCKED OUT CHANGED DISK DATA/<15><12>
 11893  050446  020107  044527  044124
 11894  050454  053440  044522  042524
 11895  050462  020123  047514  045503
 11896  050470  042105  047440  052125
 11897  050476  041440  040510  043516
 11898  050504  042105  042040  051511
 11899  050512  020113  040504  040524
 11900  050520  005015
 11901  050522  047507  042117  042040          .ASCII  /GOOD DATA GIVES WHAT WAS ON DISK BEFORE WRITE WITH WRITE LOCKED OUT/<15
 11902  050530  052101  020101  044507
 11903  050536  042526  020123  044127
 11904  050544  052101  053440  051501
 11905  050552  047440  020116  044504
 11906  050560  045523  041040  043105
 11907  050566  051117  020105  051127
 11908  050574  052111  020105  044527
 11909  050602  044124  053440  044522
 11910  050610  042524  046040  041517
 11911  050616  042513  020104  052517
 11912  050624  006524     012
 11913  050627     127  051501  040440          .ASCII  /WAS ATTEMPTED/<15><12>
 11914  050634  052124  046505  052120
 11915  050642  042105  005015
 11916  050646  042522  042503  053111          .ASCII  /RECEIVED DATA GIVES WHAT WAS READ BACK AFTER WRITE/<15><12>
 11917  050654  042105  042040  052101
 11918  050662  020101  044507  042526
 11919  050670  020123  044127  052101
 11920  050676  053440  051501  051040
 11921  050704  040505  020104  040502
 11922  050712  045503  040440  052106
 11923  050720  051105  053440  044522
 11924  050726  042524  005015
 11925  050732  044527  044124  053440          .ASCIZ  /WITH WRITE LOCKED OUT WAS ATTEMPTED/
 11926  050740  044522  042524  046040
 11927  050746  041517  042513  020104
 11928  050754  052517  020124  040527
 11929  050762  020123  052101  042524
 11930  050770  050115  042524  000104
 11931  050776  047105  041101  044514  EM44:   .ASCII  /ENABLING WRITES BY WRITE LOCK BUTTON CAUSED IMPROPER REGISTER CHANGE/<1
```

```
 11932  051004  043516  053440  044522
 11933  051012  042524  020123  054502
 11934  051020  053440  044522  042524
 11935  051026  046040  041517  020113
 11936  051034  052502  052124  047117
 11937  051042  041440  052501  042523
 11938  051050  020104  046511  051120
 11939  051056  050117  051105  051040
 11940  051064  043505  051511  042524
 11941  051072  020122  044103  047101
 11942  051100  042507  005015
 11943  051104  047507  042117  042040          .ASCII  /GOOD DATA GIVES WHAT SHOULD BE THERE/<15><12>
 11944  051112  052101  020101  044507
 11945  051120  042526  020123  044127
 11946  051126  052101  051440  047510
 11947  051134  046125  020104  042502
 11948  051142  052040  042510  042522
 11949  051150  005015
 11950  051152  042522  042503  053111          .ASCII  /RECEIVED DATA GIVES REGISTER CONTENTS AFTER WRITE LOCK BUTTON/<15><12>
 11951  051160  042105  042040  052101
 11952  051166  020101  044507  042526
 11953  051174  020123  042522  044507
 11954  051202  052123  051105  041440
 11955  051210  047117  042524  052116
 11956  051216  020123  043101  042524
 11957  051224  020122  051127  052111
 11958  051232  020105  047514  045503
 11959  051240  041040  052125  047524
 11960  051246  006516     012
 11961  051251     105  040516  046102          .ASCIZ  /ENABLED WRITES/
 11962  051256  042105  053440  044522
 11963  051264  042524  000123
 11964  051270  051124  047101  043123  EM45:   .ASCII  /TRANSFERRING ON LAST BLOCK - CYLINDER 410. - 814., SECTOR 21, /<15><12>
 11965  051276  051105  044522  043516
 11966  051304  047440  020116  040514
 11967  051312  052123  041040  047514
 11968  051320  045503  026440  041440
 11969  051326  046131  047111  042504
 11970  051334  020122  030464  027060
 11971  051342  026440  034040  032061
 11972  051350  026056  051440  041505
 11973  051356  047524  020122  030462
 11974  051364  020054  005015
 11975  051370  051124  041501  020113          .ASCII  /TRACK 18, CAUSED IMPROPER REGISTER CHANGE/<15><12>
 11976  051376  034061  020054  040503
 11977  051404  051525  042105  044440
 11978  051412  050115  047522  042520
 11979  051420  020122  042522  044507
 11980  051426  052123  051105  041440
 11981  051434  040510  043516  006505
 11982  051442     012
 11983  051443     107  047517  020104          .ASCII  /GOOD DATA GIVES WHAT SHOULD BE THERE/<15><12>
 11984  051450  040504  040524  043440
 11985  051456  053111  051505  053440
 11986  051464  040510  020124  044123
 11987  051472  052517  042114  041040
```

```
 11988  051500  020105  044124  051105
 11989  051506  006505     012
 11990  051511     122  041505  044505          .ASCIZ  /RECEIVED DATA GIVES REGISTEP CONTENTS AFTER TRANSFER/
 11991  051516  042526  020104  040504
 11992  051524  040524  043440  053111
 11993  051532  051505  051040  043505
 11994  051540  051511  042524  020122
 11995  051546  047503  052116  047105
 11996  051554  051524  040440  052106
 11997  051562  051105  052040  040522
 11998  051570  051516  042506  000122
 11999  051576  040504  040524  051040  EM46:   .ASCII  /DATA READ FROM LAST BLOCK - CYLINDER 410. - 814., SECTOR 21,/<15><12>
 12000  051604  040505  020104  051106
 12001  051612  046517  046040  051501
 12002  051620  020124  046102  041517
 12003  051626  020113  020055  054503
 12004  051634  044514  042116  051105
 12005  051642  032040  030061  020056
 12006  051650  020055  030470  027064
 12007  051656  020054  042523  052103
 12008  051664  051117  031040  026061
 12009  051672  005015
 12010  051674  051124  041501  020113          .ASCIZ  /TRACK 18, IS IN ERROR/
 12011  051702  034061  020054  051511
 12012  051710  044440  020116  051105
 12013  051716  047522  000122
 12014  051722  051124  047101  043123  EM47:   .ASCII  /TRANSFERRING DATA FROM NONEXISTANT SECTOR CAUSED IMPROPER /<15><12>
 12015  051730  051105  044522  043516
 12016  051736  042040  052101  020101
 12017  051744  051106  046517  047040
 12018  051752  047117  054105  051511
 12019  051760  040524  052116  051440
 12020  051766  041505  047524  020122
 12021  051774  040503  051525  042105
 12022  052002  044440  050115  047522
 12023  052010  042520  020122  005015
 12024  052016  042522  044507  052123          .ASCII  /REGISTER CHANGE, GOOD DATA GIVES WHAT SHOULD BE THERE/<15><12>
 12025  052024  051105  041440  040510
 12026  052032  043516  026105  043440
 12027  052040  047517  020104  040504
 12028  052046  040524  043440  053111
 12029  052054  051505  053440  040510
 12030  052062  020124  044123  052517
 12031  052070  042114  041040  020105
 12032  052076  044124  051105  006505
 12033  052104     012
 12034  052105     122  041505  044505          .ASCIZ  /RECEIVED DATA GIVES REGISTER CONTENTS AFTER ATTEMPTED TRANSFER/
 12035  052112  042526  020104  040504
 12036  052120  040524  043440  053111
 12037  052126  051505  051040  043505
 12038  052134  051511  042524  020122
 12039  052142  047503  052116  047105
 12040  052150  051524  040440  052106
 12041  052156  051105  040440  052124
 12042  052164  046505  052120  042105
 12043  052172  052040  040522  051516
```

```
12044  052200  042506  000122
12045  052204  051124  047101  043123  EM50:   .ASCII  /TRANSFERRING FROM NONEXISTANT SECTOR CAUSED DATA ERROR/<15><12>
12046  052212  051105  044522  043516
12047  052220  043040  047522  020115
12048  052226  047516  042516  044530
12049  052234  052123  047101  020124
12050  052242  042523  052103  051117
12051  052250  041440  052501  042523
12052  052256  020104  040504  040524
12053  052264  042440  051122  051117
12054  052272  005015
12055  052274  047507  042117  042040          .ASCII  /GOOD DATA GIVES WHAT SHOULD BE THERE/<15><12>
12056  052302  052101  020101  044507
12057  052310  042526  020123  044127
12058  052316  052101  051440  047510
12059  052324  046125  020104  042502
12060  052332  052040  042510  042522
12061  052340  005015
12062  052342  040502  020104  040504          .ASCIZ  /BAD DATA GIVES WHAT WAS IN BUFFER AFTER TRANSFER/
12063  052350  040524  043440  053111
12064  052356  051505  053440  040510
12065  052364  020124  040527  020123
12066  052372  047111  041040  043125
12067  052400  042506  020122  043101
12068  052406  042524  020122  051124
12069  052414  047101  043123  051105
12070  052422     000
12071
12072  052423     107  053111  047111  EM51:   .ASCII  /GIVING ILLEGAL FUNCTION CAUSED IMPROPER REGISTER CHANGE/<15><12>
12073  052430  020107  046111  042514
12074  052436  040507  020114  052506
12075  052444  041516  044524  047117
12076  052452  041440  052501  042523
12077  052460  020104  046511  051120
12078  052466  050117  051105  051040
12079  052474  043505  051511  042524
12080  052502  020122  044103  047101
12081  052510  042507  005015
12082  052514  047507  042117  042040          .ASCII  /GOOD DATA GIVES WHAT SHOULD BE THERE/<15><12>
12083  052522  052101  020101  044507
12084  052530  042526  020123  044127
12085  052536  052101  051440  047510
12086  052544  046125  020104  042502
12087  052552  052040  042510  042522
12088  052560  005015
12089  052562  042522  042503  053111          .ASCIZ  /RECEIVED DATA GIVES REGISTER CONTENTS AFTER ILLEGAL FUNCTION IS GIVEN/
12090  052570  042105  042040  052101
12091  052576  020101  044507  042526
12092  052604  020123  042522  044507
12093  052612  052123  051105  041440
12094  052620  047117  042524  052116
12095  052626  020123  043101  042524
12096  052634  020122  046111  042514
12097  052642  040507  020114  052506
12098  052650  041516  044524  047117
12099  052656  044440  020123  044507
```

```
 12100  052664  042526  000116
 12101  052670  051127  052111  020105  EM52:   .ASCII  /WRITE DATA ON NONEXISTANT SECTOR CAUSED IMPROPER REGISTER CHANGE/<15><1
 12102  052676  040504  040524  047440
 12103  052704  020116  047516  042516
 12104  052712  044530  052123  047101
 12105  052720  020124  042523  052103
 12106  052726  051117  041440  052501
 12107  052734  042523  020104  046511
 12108  052742  051120  050117  051105
 12109  052750  051040  043505  051511
 12110  052756  042524  020122  044103
 12111  052764  047101  042507  005015
 12112  052772  047507  042117  042040          .ASCII  /GOOD DATA GIVES WHAT SHOULD BE THERE/<15><12>
 12113  053000  052101  020101  044507
 12114  053006  042526  020123  044127
 12115  053014  052101  051440  047510
 12116  053022  046125  020104  042502
 12117  053030  052040  042510  042522
 12118  053036  005015
 12119  053040  042522  042503  053111          .ASCIZ  /RECEIVED DATA GIVES REGISTER CONTENTS AFTER ATTEMPTED WRITE DATA/
 12120  053046  042105  042040  052101
 12121  053054  020101  044507  042526
 12122  053062  020123  042522  044507
 12123  053070  052123  051105  041440
 12124  053076  047117  042524  052116
 12125  053104  020123  043101  042524
 12126  053112  020122  052101  042524
 12127  053120  050115  042524  020104
 12128  053126  051127  052111  020105
 12129  053134  040504  040524     000
 12130  053141     122  040505  020104  EM53:   .ASCIZ  /READ HEADER AND DATA AFTER A SEARCH CAUSED DATA ERROR/
 12131  053146  042510  042101  051105
 12132  053154  040440  042116  042040
 12133  053162  052101  020101  043101
 12134  053170  042524  020122  020101
 12135  053176  042523  051101  044103
 12136  053204  041440  052501  042523
 12137  053212  020104  040504  040524
 12138  053220  042440  051122  051117
 12139  053226     000
 12140  053227     101  052124  046505  EM54:   .ASCII  /ATTEMPTING COMMAND WITH INVALID ADDRESS CAUSED IMPROPER REGISTER CHANGE
 12141  053234  052120  047111  020107
 12142  053242  047503  046515  047101
 12143  053250  020104  044527  044124
 12144  053256  044440  053116  046101
 12145  053264  042111  040440  042104
 12146  053272  042522  051523  041440
 12147  053300  052501  042523  020104
 12148  053306  046511  051120  050117
 12149  053314  051105  051040  043505
 12150  053322  051511  042524  020122
 12151  053330  044103  047101  042507
 12152  053336  005015
 12153  053340  047507  042117  042040          .ASCII  /GOOD DATA GIVES WHAT SHOULD BE THERE/<15><12>
 12154  053346  052101  020101  044507
 12155  053354  042526  020123  044127
```

```
12156  053362  052101  051440  047510
12157  053370  046125  020104  042502
12158  053376  052040  042510  042522
12159  053404  005015
12160  053406  042522  042503  053111          .ASCIZ  /RECEIVED DATA GIVES REGISTER CONTENTS AFTER OPERATION/
12161  053414  042105  042040  052101
12162  053422  020101  044507  042526
12163  053430  020123  042522  044507
12164  053436  052123  051105  041440
12165  053444  047117  042524  052116
12166  053452  020123  043101  042524
12167  053460  020122  050117  051105
12168  053466  052101  047511  000116
12169  053474  051127  052111  047111  EM55:   .ASCII  /WRITING OR READING WITH EXPECTED ADDRESS OVERFLOW ERROR/<15><12>
12170  053502  020107  051117  051040
12171  053510  040505  044504  043516
12172  053516  053440  052111  020110
12173  053524  054105  042520  052103
12174  053532  042105  040440  042104
12175  053540  042522  051523  047440
12176  053546  042526  043122  047514
12177  053554  020127  051105  047522
12178  053562  006522     012
12179  053565     103  052501  042523          .ASCII  /CAUSED IMPROPER REGISTER CHANGE/<15><12>
12180  053572  020104  046511  051120
12181  053600  050117  051105  051040
12182  053606  043505  051511  042524
12183  053614  020122  044103  047101
12184  053622  042507  005015
12185  053626  047507  042117  042040          .ASCII  /GOOD DATA GIVES WHAT SHOULD BE THERE/<15><12>
12186  053634  052101  020101  044507
12187  053642  042526  020123  044127
12188  053650  052101  051440  047510
12189  053656  046125  020104  042502
12190  053664  052040  042510  042522
12191  053672  005015
12192  053674  042522  042503  053111          .ASCIZ  /RECEIVED DATA GIVES REGISTER CONTENTS AFTER OPERATION/
12193  053702  042105  042040  052101
12194  053710  020101  044507  042526
12195  053716  020123  042522  044507
12196  053724  052123  051105  041440
12197  053732  047117  042524  052116
12198  053740  020123  043101  042524
12199  053746  020122  050117  051105
12200  053754  052101  047511  000116
12201  053762  040504  040524  051040  EM56:   .ASCII  /DATA READ WITH AN EXPECTED ADDRESS OVERFLOW ERROR IS INCORRECT/<15><12>
12202  053770  040505  020104  044527
12203  053776  044124  040440  020116
12204  054004  054105  042520  052103
12205  054012  042105  040440  042104
12206  054020  042522  051523  047440
12207  054026  042526  043122  047514
12208  054034  020127  051105  047522
12209  054042  020122  051511  044440
12210  054050  041516  051117  042522
12211  054056  052103  005015
```

```
12212  054062  047527  042122  047040          .ASCII  /WORD NO. 1 TO 260 SHOULD BE READ, WORD NO 261 TO 266 SHOULD/<15><12>
12213  054070  027117  030440  052040
12214  054076  020117  033062  020060
12215  054104  044123  052517  042114
12216  054112  041040  020105  042522
12217  054120  042101  020054  047527
12218  054126  042122  047040  020117
12219  054134  033062  020061  047524
12220  054142  031040  033066  051440
12221  054150  047510  046125  006504
12222  054156     012
12223  054157     102  020105  044103          .ASCIZ  /BE CHANGED/
12224  054164  047101  042507  000104
12225  054172  052101  042524  050115  EM57:   .ASCII  /ATTEMPTING DATA COMMAND WITH WRONG FORMAT BIT CAUSED/<15><12>
12226  054200  044524  043516  042040
12227  054206  052101  020101  047503
12228  054214  046515  047101  020104
12229  054222  044527  044124  053440
12230  054230  047522  043516  043040
12231  054236  051117  040515  020124
12232  054244  044502  020124  040503
12233  054252  051525  042105  005015
12234  054260  046511  051120  050117          .ASCII  /IMPROPER REGISTER CHANGE/<15><12>
12235  054266  051105  051040  043505
12236  054274  051511  042524  020122
12237  054302  044103  047101  042507
12238  054310  005015
12239  054312  047507  042117  042040          .ASCII  /GOOD DATA GIVES WHAT SHOULD BE THERE/<15><12>
12240  054320  052101  020101  044507
12241  054326  042526  020123  044127
12242  054334  052101  051440  047510
12243  054342  046125  020104  042502
12244  054350  052040  042510  042522
12245  054356  005015
12246  054360  042522  042503  053111          .ASCIZ  /RECEIVED DATA GIVES REGISTER CONTENTS AFTER ATTEMPTED DATA TRANSFER/
12247  054366  042105  042040  052101
12248  054374  020101  044507  042526
12249  054402  020123  042522  044507
12250  054410  052123  051105  041440
12251  054416  047117  042524  052116
12252  054424  020123  043101  042524
12253  054432  020122  052101  042524
12254  054440  050115  042524  020104
12255  054446  040504  040524  052040
12256  054454  040522  051516  042506
12257  054462  000122
12258  054464  052101  042524  050115  EM60:   .ASCII  /ATTEMPTING TO MODIFY REGISTER DURING AN OPERATION CAUSED IMPROPER/<15><
12259  054472  044524  043516  052040
12260  054500  020117  047515  044504
12261  054506  054506  051040  043505
12262  054514  051511  042524  020122
12263  054522  052504  044522  043516
12264  054530  040440  020116  050117
12265  054536  051105  052101  047511
12266  054544  020116  040503  051525
12267  054552  042105  044440  050115
```

```
12268  054560  047522  042520  006522
12269  054566     012
12270  054567     122  043505  051511          .ASCII  /REGISTER CHANGE. GOOD DATA GIVES WHAT SHOULD BE THERE/<15><12>
12271  054574  042524  020122  044103
12272  054602  047101  042507  020056
12273  054610  047507  042117  042040
12274  054616  052101  020101  044507
12275  054624  042526  020123  044127
12276  054632  052101  051440  047510
12277  054640  046125  020104  042502
12278  054646  052040  042510  042522
12279  054654  005015
12280  054656  042522  042503  053111          .ASCII  /RECEIVED DATA GIVES REGISTER CONTENTS AFTER OPERATION WAS ATTEMPTED/<15
12281  054664  042105  042040  052101
12282  054672  020101  044507  042526
12283  054700  020123  042522  044507
12284  054706  052123  051105  041440
12285  054714  047117  042524  052116
12286  054722  020123  043101  042524
12287  054730  020122  050117  051105
12288  054736  052101  047511  020116
12289  054744  040527  020123  052101
12290  054752  042524  050115  042524
12291  054760  006504     012
12292  054763     115  042117  044506          .ASCIZ  /MODFING REG GIVES ADDRESS OF REGISTER BEING MODIFIED WHICH CAUSED ERROR
12293  054770  043516  051040  043505
12294  054776  043440  053111  051505
12295  055004  040440  042104  042522
12296  055012  051523  047440  020106
12297  055020  042522  044507  052123
12298  055026  051105  041040  044505
12299  055034  043516  046440  042117
12300  055042  043111  042511  020104
12301  055050  044127  041511  020110
12302  055056  040503  051525  042105
12303  055064  042440  051122  051117
12304  055072     000
12305
12306  055073     104  053105  041511  EM61:   .ASCIZ  /DEVICE NOT AVAILABLE BEFORE COMMAND WAS TO BE GIVEN/
12307  055100  020105  047516  020124
12308  055106  053101  044501  040514
12309  055114  046102  020105  042502
12310  055122  047506  042522  041440
12311  055130  046517  040515  042116
12312  055136  053440  051501  052040
12313  055144  020117  042502  043440
12314  055152  053111  047105     000
12315  055157     122  042110  030523  EM63:   .ASCIZ  /RHDS1 CONTENTS DURING COMMAND WAS IN ERROR/
12316  055164  041440  047117  042524
12317  055172  052116  020123  052504
12318  055200  044522  043516  041440
12319  055206  046517  040515  042116
12320  055214  053440  051501  044440
12321  055222  020116  051105  047522
12322  055230  000122
12323  055232  042522  040503  044514  EM64:   .ASCII  /RECALIBRATE COMMAND CAUSED IMPROPER REGISTER CHANGE/<15><12>
```

```
12324  055240  051102  052101  020105
12325  055246  047503  046515  047101
12326  055254  020104  040503  051525
12327  055262  042105  044440  050115
12328  055270  047522  042520  020122
12329  055276  042522  044507  052123
12330  055304  051105  041440  040510
12331  055312  043516  006505     012
12332  055317     107  047517  020104          .ASCII  /GOOD DATA GIVES WHAT SHOULD BE THERE/<15><12>
12333  055324  040504  040524  043440
12334  055332  053111  051505  053440
12335  055340  040510  020124  044123
12336  055346  052517  042114  041040
12337  055354  020105  044124  051105
12338  055362  006505     012
12339  055365     122  041505  044505          .ASCIZ  /RECEIVED DATA GIVES REGISTER CONTENTS AFTER COMMAND/
12340  055372  042526  020104  040504
12341  055400  040524  043440  053111
12342  055406  051505  051040  043505
12343  055414  051511  042524  020122
12344  055422  047503  052116  047105
12345  055430  051524  040440  052106
12346  055436  051105  041440  046517
12347  055444  040515  042116     000
12348  055451     111  052116  051105  EM65:   .ASCIZ  /INTERRUPT FAILING/
12349  055456  052522  052120  043040
12350  055464  044501  044514  043516
12351  055472     000
12352  055473     110  040505  042504  EM66:   .ASCII  /HEADER AND DATA COMMAND FOR HEAD SELECTION TEST/<15><12>
12353  055500  020122  047101  020104
12354  055506  040504  040524  041440
12355  055514  046517  040515  042116
12356  055522  043040  051117  044040
12357  055530  040505  020104  042523
12358  055536  042514  052103  047511
12359  055544  020116  042524  052123
12360  055552  005015
12361  055554  040503  051525  042105          .ASCII  /CAUSED ERROR/<15><12>
12362  055562  042440  051122  051117
12363  055570  005015
12364  055572  044122  051504  020124          .ASCII  /RHDST GIVES WHAT TRACK WAS BEING WRITTEN OR READ/<15><12>
12365  055600  044507  042526  020123
12366  055606  044127  052101  052040
12367  055614  040522  045503  053440
12368  055622  051501  041040  044505
12369  055630  043516  053440  044522
12370  055636  052124  047105  047440
12371  055644  020122  042522  042101
12372  055652  005015
12373  055654  047117  041440  046131          .ASCIZ  /ON CYLINDER 0, SECTOR 0/
12374  055662  047111  042504  020122
12375  055670  026060  051440  041505
12376  055676  047524  020122  000060
12377  055704  042522  042101  044040  EM67:   .ASCII  /READ HEADER AND DATA ERROR IN HEAD SELECTION TEST/<12><15>
12378  055712  040505  042504  020122
12379  055720  047101  020104  040504
```

```
12380  055726  040524  042440  051122
12381  055734  051117  044440  020116
12382  055742  042510  042101  051440
12383  055750  046105  041505  044524
12384  055756  047117  052040  051505
12385  055764  005124     015
12386  055767     106  051111  052123          .ASCII  /FIRST FOUR WORD NUMBERS ARE HEADER/<12><15>
12387  055774  043040  052517  020122
12388  056002  047527  042122  047040
12389  056010  046525  042502  051522
12390  056016  040440  042522  044040
12391  056024  040505  042504  005122
12392  056032     015
12393  056033     127  051117  020104          .ASCII  /WORD NUMBERS 5 TO 260 ARE DATA WORDS/<12><15>
12394  056040  052516  041115  051105
12395  056046  020123  020065  047524
12396  056054  031040  030066  040440
12397  056062  042522  042040  052101
12398  056070  020101  047527  042122
12399  056076  005123     015
12400  056101     111  020116  040504          .ASCIZ  /IN DATA WORDS BITS 4,5,6,7,8 GIVE TRACK NUMBER/
12401  056106  040524  053440  051117
12402  056114  051504  041040  052111
12403  056122  020123  026064  026065
12404  056130  026066  026067  020070
12405  056136  044507  042526  052040
12406  056144  040522  045503  047040
12407  056152  046525  042502  000122
12408
12409  056160  042522  042101  044040  EM70:   .ASCII  /READ HEADER AND DATA ERROR IN/<15><12>
12410  056166  040505  042504  020122
12411  056174  047101  020104  040504
12412  056202  040524  042440  051122
12413  056210  051117  044440  006516
12414  056216     012
12415  056217     104  043111  042506          .ASCII  /DIFFERENCE LINE TEST/<15><12>
12416  056224  042522  041516  020105
12417  056232  044514  042516  052040
12418  056240  051505  006524     012
12419  056245     127  051117  020104          .ASCII  /WORD NOS 1-4 GIVE HEADER/<15><12>
12420  056252  047516  020123  026461
12421  056260  020064  044507  042526
12422  056266  044040  040505  042504
12423  056274  006522     012
12424  056277     127  051117  020104          .ASCIZ  /WORD NOS 5-260 GIVE DATA WHICH IS THE CYLINDER ADDRESS/
12425  056304  047516  020123  026465
12426  056312  033062  020060  044507
12427  056320  042526  042040  052101
12428  056326  020101  044127  041511
12429  056334  020110  051511  052040
12430  056342  042510  041440  046131
12431  056350  047111  042504  020122
12432  056356  042101  051104  051505
12433  056364  000123
12434  056366  047506  041522  047111  EM71:   .ASCII  /FORCING OPI BY 3 INDEX PULSES/<15><12>
12435  056374  020107  050117  020111
```

```
12436  056402  054502  031440  044440
12437  056410  042116  054105  050040
12438  056416  046125  042523  006523
12439  056424     012
12440  056425     103  052501  042523          .ASCII  /CAUSED IMPROPER REGISTER CHANGE/<15><12>
12441  056432  020104  046511  051120
12442  056440  050117  051105  051040
12443  056446  043505  051511  042524
12444  056454  020122  044103  047101
12445  056462  042507  005015
12446  056466  047507  042117  042040          .ASCII  /GOOD DATA GIVES WHAT SHOULD BE THERE/<15><12>
12447  056474  052101  020101  044507
12448  056502  042526  020123  044127
12449  056510  052101  051440  047510
12450  056516  046125  020104  042502
12451  056524  052040  042510  042522
12452  056532  005015
12453  056534  042522  042503  053111          .ASCIZ  /RECEIVED DATA GIVES REGISTER CONTENTS AFTER 3 INDEX PULSES/
12454  056542  042105  042040  052101
12455  056550  020101  044507  042526
12456  056556  020123  042522  044507
12457  056564  052123  051105  041440
12458  056572  047117  042524  052116
12459  056600  020123  043101  042524
12460  056606  020122  020063  047111
12461  056614  042504  020130  052520
12462  056622  051514  051505     000
12463  056627     124  042510  042522  EM72:   .ASCII  /THERE WAS A SETUP ERROR DURING MULTIPLE WRITE/<15><12>
12464  056634  053440  051501  040440
12465  056642  051440  052105  050125
12466  056650  042440  051122  051117
12467  056656  042040  051125  047111
12468  056664  020107  052515  052114
12469  056672  050111  042514  053440
12470  056700  044522  042524  005015
12471  056706  042510  042101  051105          .ASCII  /HEADER AND DATA COMMANDS RESULTING IN AN ABORT/<15><12>
12472  056714  040440  042116  042040
12473  056722  052101  020101  047503
12474  056730  046515  047101  051504
12475  056736  051040  051505  046125
12476  056744  044524  043516  044440
12477  056752  020116  047101  040440
12478  056760  047502  052122  005015
12479  056766  043117  052040  044510          .ASCII  /OF THIS 'OPI' TEST./<15><12><15><12>
12480  056774  020123  047447  044520
12481  057002  020047  042524  052123
12482  057010  006456  006412     012
12483  057015     124  020117  051124          .ASCIZ  /TO TROUBLE SHOOT SETUP ERROR, LOOP ON THIS TEST/
12484  057022  052517  046102  020105
12485  057030  044123  047517  020124
12486  057036  042523  052524  020120
12487  057044  051105  047522  026122
12488  057052  046040  047517  020120
12489  057060  047117  052040  044510
12490  057066  020123  042524  052123
12491  057074     000
```

```
12492  057075     122  040505  020104  EM73:   .ASCII  /READ HEADER AND DATA FOR 11960 WORDS /<15><12>
12493  057102  042510  042101  051105
12494  057110  040440  042116  042040
12495  057116  052101  020101  047506
12496  057124  020122  030461  033071
12497  057132  020060  047527  042122
12498  057140  020123  005015
12499  057144  044124  052101  044440          .ASCII  /THAT IS 46 SECTORS /<15><12>
12500  057152  020123  033064  051440
12501  057160  041505  047524  051522
12502  057166  006440     012
12503  057171     124  040510  020124          .ASCIZ  /THAT IS OVER 3 INDEX PULSES CAUSED AN ERROR/
12504  057176  051511  047440  042526
12505  057204  020122  020063  047111
12506  057212  042504  020130  052520
12507  057220  051514  051505  041440
12508  057226  052501  042523  020104
12509  057234  047101  042440  051122
12510  057242  051117     000
12511  057245     122  040505  020104  EM74:   .ASCII  /READ HEADER AND DATA FOR 11960 WORDS /<15><12>
12512  057252  042510  042101  051105
12513  057260  040440  042116  042040
12514  057266  052101  020101  047506
12515  057274  020122  030461  033071
12516  057302  020060  047527  042122
12517  057310  020123  005015
12518  057314  044124  052101  044440          .ASCII  /THAT IS 46 SECTORS, THAT IS OVER 3 INDEX /<15><12>
12519  057322  020123  033064  051440
12520  057330  041505  047524  051522
12521  057336  020054  044124  052101
12522  057344  044440  020123  053117
12523  057352  051105  031440  044440
12524  057360  042116  054105  006440
12525  057366     012
12526  057367     120  046125  042523          .ASCIZ  /PULSES CAUSED OPI TO SET/
12527  057374  020123  040503  051525
12528  057402  042105  047440  044520
12529  057410  052040  020117  042523
12530  057416  000124
12531  057420  050040  047522  051107  NOUSE:  .ASCIZ  / PROGRAMMABLE-DRIVE WILL NOT BE USED/
12532  057426  046501  040515  046102
12533  057434  026505  051104  053111
12534  057442  020105  044527  046114
12535  057450  047040  052117  041040
12536  057456  020105  051525  042105
12537  057464     000
```

```
12538
12539  057465     106  052101  046101  CPHALT: .ASCII  /FATAL ERROR - SEE DOCUMENT LISTING/<15><12>
12540  057472  042440  051122  051117
12541  057500  026440  051440  042505
12542  057506  042040  041517  046525
12543  057514  047105  020124  044514
12544  057522  052123  047111  006507
12545  057530     012
12546  057531     040  005015  177607          .ASCII  / /<15><12><207><377><377><207><377><377><207><377><377>
12547  057536  103777  177777  177607
12548  057544     377
12549  057545     124  042510  041440          .ASCII  /THE CONTROLLER OR DEVICE HAS GONE OFFLINE, LOST/<15><12>
12550  057552  047117  051124  046117
12551  057560  042514  020122  051117
12552  057566  042040  053105  041511
12553  057574  020105  040510  020123
12554  057602  047507  042516  047440
12555  057610  043106  044514  042516
12556  057616  020054  047514  052123
12557  057624  005015
12558  057626  051047  040505  054504          .ASCII  /'READY', BECOME UNAVAILABLE, OR HAS STATUS BITS/<15><12>
12559  057634  026047  041040  041505
12560  057642  046517  020105  047125
12561  057650  053101  044501  040514
12562  057656  046102  026105  047440
12563  057664  020122  040510  020123
12564  057672  052123  052101  051525
12565  057700  041040  052111  006523
12566  057706     012
12567  057707     127  044510  044103          .ASCIZ  /WHICH CANNOT BE CLEARED/
12568  057714  041440  047101  047516
12569  057722  020124  042502  041440
12570  057730  042514  051101  042105
12571  057736     000
12572
12573
12574
12575  057737     120  020103  020040  DH1:    .ASCII  /PC      TEST    WAIT    BIT     REG     REG     RHCS1/<15><12>
12576  057744  020040  052040  051505
12577  057752  020124  020040  053440
12578  057760  044501  020124  020040
12579  057766  041040  052111  020040
12580  057774  020040  051040  043505
12581  060002  020040  020040  051040
12582  060010  043505  020040  020040
12583  060016  051040  041510  030523
12584  060024  005015
12585  060026  020040  020040  020040          .ASCIZ  /       NO      PC      EXPCTD  ADDRESS CONTENT CONTENT       /
12586  060034  020040  047516  020040
12587  060042  020040  020040  041520
12588  060050  020040  020040  020040
12589  060056  054105  041520  042124
12590  060064  020040  042101  051104
12591  060072  051505  020123  047503
12592  060100  052116  047105  020124
12593  060106  047503  052116  047105
```

```
12594  060114  004524     000
12595  060117     120  020103  020040  DH4:    .ASCII  /PC      TEST    WAIT    BIT     REG     TIME IN/<15><12>
12596  060124  020040  052040  051505
12597  060132  020124  020040  053440
12598  060140  044501  020124  020040
12599  060146  041040  052111  020040
12600  060154  020040  051040  043505
12601  060162  020040  020040  052040
12602  060170  046511  020105  047111
12603  060176  005015
12604  060200  020040  020040  020040          .ASCIZ  /       NO      PC      EXPCTD  ADDRESS 10 MSEC/
12605  060206  020040  047516  020040
12606  060214  020040  020040  041520
12607  060222  020040  020040  020040
12608  060230  054105  041520  042124
12609  060236  020040  042101  051104
12610  060244  051505  020123  030061
12611  060252  046440  042523  000103
12612  060260  041520  020040  020040  DH5:    .ASCII  /PC      TEST    REG     GOOD    RECEIVED/<15><12>
12613  060266  020040  042524  052123
12614  060274  020040  020040  042522
12615  060302  020107  020040  020040
12616  060310  047507  042117  020040
12617  060316  020040  042522  042503
12618  060324  053111  042105  005015
12619  060332  020040  020040  020040          .ASCIZ  /       NO      ADDRESS DATA    DATA/
12620  060340  020040  047516  020040
12621  060346  020040  020040  042101
12622  060354  051104  051505  020123
12623  060362  040504  040524  020040
12624  060370  020040  040504  040524
12625  060376     000
12626  060377     120  020103  020040  DH6:    .ASCII  /PC      TEST    REG     RECEIVED/<15><12>
12627  060404  020040  052040  051505
12628  060412  020124  020040  051040
12629  060420  043505  020040  020040
12630  060426  051040  041505  044505
12631  060434  042526  006504     012
12632  060441     040  020040  020040          .ASCIZ  /       NO      ADDRESS DATA/
12633  060446  020040  047040  020117
12634  060454  020040  020040  040440
12635  060462  042104  042522  051523
12636  060470  042040  052101  000101
12637  060476  041520  020040  020040  DH7:    .ASCIZ  /PC      TEST    REG     ADDRESS/
12638  060504  020040  042524  052123
12639  060512  020040  020040  042522
12640  060520  020107  020040  020040
12641  060526  042101  051104  051505
12642  060534  000123
12643
12644  060536  041520  020040  020040  DH10:   .ASCII  /PC      TEST    FAILING CONTENT CONTENT CONTENT CONTENT/<15><12>
12645  060544  020040  042524  052123
12646  060552  020040  020040  040506
12647  060560  046111  047111  020107
12648  060566  047503  052116  047105
12649  060574  020124  047503  052116
```

```
12650  060602  047105  020124  047503
12651  060610  052116  047105  020124
12652  060616  047503  052116  047105
12653  060624  006524    012
12654  060627    040   020040  020040          .ASCIZ  /       NO      REG.    RHCS1   RHCS2   RHDS1   RHER1/
12655  060634  020040  047040  020117
12656  060642  020040  020040  051040
12657  060650  043505  020056  020040
12658  060656  051040  041510  030523
12659  060664  020040  051040  041510
12660  060672  031123  020040  051040
12661  060700  042110  030523  020040
12662  060706  051040  042510  030522
12663  060714    000
12664
12665  060715    120   020103  020040  DH26:   .ASCII  /PC     TEST    CONT OF CONT OF CONT OF CONT OF CONT OF CONT OF/<15><12
12666  060722  020040  052040  051505
12667  060730  020124  020040  041440
12668  060736  047117  020124  043117
12669  060744  041440  047117  020124
12670  060752  043117  041440  047117
12671  060760  020124  043117  041440
12672  060766  047117  020124  043117
12673  060774  041440  047117  020124
12674  061002  043117  041440  047117
12675  061010  020124  043117  005015
12676  061016  020040  020040  020040          .ASCIZ  /       NO      RHCS1   RHCS2   RHDS1   RHER1   RHER2   RHER3  /
12677  061024  020040  047516  020040
12678  061032  020040  020040  044122
12679  061040  051503  020061  020040
12680  061046  044122  051503  020062
12681  061054  020040  044122  051504
12682  061062  020061  020040  044122
12683  061070  051105  020061  020040
12684  061076  044122  051105  020062
12685  061104  020040  044122  051105
12686  061112  004463    000
12687
12688  061115    120   020103  020040  DH30:   .ASCII  /PC     TEST    WORD    GOOD    BAD/<15><12>
12689  061122  020040  052040  051505
12690  061130  020124  020040  053440
12691  061136  051117  020104  020040
12692  061144  043440  047517  020104
12693  061152  020040  041040  042101
12694  061160  005015
12695  061162  020040  020040  020040          .ASCIZ  /       NO      NO      DATA    DATA/
12696  061170  020040  047516  020040
12697  061176  020040  020040  047516
12698  061204  020040  020040  020040
12699  061212  040504  040524  020040
12700  061220  020040  040504  040524
12701  061226    000
12702
12703  061227    120   020103  020040  DH51:   .ASCII  /PC     TEST    REG     GOOD    RECVD   ILLEGL/<15><12>
12704  061234  020040  052040  051505
12705  061242  020124  020040  051040
```

```
12706  061250  043505  020040  020040
12707  061256  043440  047517  020104
12708  061264  020040  051040  041505
12709  061272  042126  020040  044440
12710  061300  046114  043505  006514
12711  061306     012
12712  061307     040  020040  020040          .ASCIZ  /       NO      ADDRESS DATA    DATA    FUNCTN/
12713  061314  020040  047040  020117
12714  061322  020040  020040  040440
12715  061330  042104  042522  051523
12716  061336  042040  052101  020101
12717  061344  020040  042040  052101
12718  061352  020101  020040  043040
12719  061360  047125  052103  000116
12720
12721  061366  041520  020040  020040  DH60:   .ASCII  /PC     TEST    REG     GOOD    RECVD   MODFING/<15><12>
12722  061374  020040  042524  052123
12723  061402  020040  020040  042522
12724  061410  020107  020040  020040
12725  061416  047507  042117  020040
12726  061424  020040  042522  053103
12727  061432  020104  020040  047515
12728  061440  043104  047111  006507
12729  061446     012
12730  061447     040  020040  020040          .ASCIZ  /       NO      ADDRESS DATA    DATA    REG/
12731  061454  020040  047040  020117
12732  061462  020040  020040  040440
12733  061470  042104  042522  051523
12734  061476  042040  052101  020101
12735  061504  020040  042040  052101
12736  061512  020101  020040  051040
12737  061520  043505     000
12738  061523     120  020103  020040  DH61:   .ASCII  /PC     TEST    PC OF   RHDS1/<15><12>
12739  061530  020040  052040  051505
12740  061536  020124  020040  050040
12741  061544  020103  043117  020011
12742  061552  044122  051504  006461
12743  061560     012
12744  061561     040  020040  020040          .ASCIZ  /       NO      JSR     WAS/
12745  061566  020040  047040  020117
12746  061574  020040  020040  045040
12747  061602  051123  020040  020040
12748  061610  053440  051501     000
12749  061615     120  020103  020040  DH62:   .ASCII  /PC     PC OF   RHCS1/<15><12>
12750  061622  020040  050040  020103
12751  061630  043117  020040  051040
12752  061636  041510  030523  005015
12753  061644  020040  020040  020040          .ASCIZ  /       JSR     WAS/
12754  061652  020040  051512  020122
12755  061660  020040  020040  040527
12756  061666  000123
12757  061670  041520  020040  020040  DH65:   .ASCII  /PC     TEST    CONT    CONT    CONT/<15><12>
12758  061676  020040  042524  052123
12759  061704  020040  020040  047503
12760  061712  052116  020040  020040
12761  061720  047503  052116  020040
```

```
12762  061726  020040  047503  052116
12763  061734  005015
12764  061736  020040  020040  020040          .ASCIZ  /       NO      RHCS1   RHAS    RHDS1/
12765  061744  020040  047516  020040
12766  061752  020040  020040  044122
12767  061760  051503  020061  020040
12768  061766  044122  051501  020040
12769  061774  020040  044122  051504
12770  062002  000061
12771  062004  041520  020040  020040  DH66:   .ASCIZ  /PC      TEST    RHDST   RHER1   RHER2   RHER3   RHCS1   RHCS2/
12772  062012  020040  042524  052123
12773  062020  020040  020040  044122
12774  062026  051504  020124  020040
12775  062034  044122  051105  020061
12776  062042  020040  044122  051105
12777  062050  020062  020040  044122
12778  062056  051105  020063  020040
12779  062064  044122  051503  020061
12780  062072  020040  044122  051503
12781  062100  000062
12782
12783  062102  041520  020040  020040  DH72:   .ASCIZ  /PC      TEST    RHCSI   RHCS2   RHDS1   RHDST   RHCA    RHER1   RHWC/
12784  062110  020040  042524  052123
12785  062116  020040  020040  044122
12786  062124  051503  020111  020040
12787  062132  044122  051503  020062
12788  062140  020040  044122  051504
12789  062146  020061  020040  044122
12790  062154  051504  020124  020040
12791  062162  044122  040503  020040
12792  062170  020040  044122  051105
12793  062176  020061  020040  044122
12794  062204  041527     000
12795
12796
12797          062210                   .EVEN
12798
12799  062210  001116  004504  033544  DT1:    .WORD   $ERRPC,TSTNM,WAITPC,WAITBT,WAITRE,$BDDAT,CS1,0
12800  062216  033550  033546  001126
12801  062224  002262  000000
12802  062230  001116  004504  004504  DT4:    .WORD   $ERRPC,TSTNM,TSTNM,WAITPC,WAITBT,WAITRE,$BDDAT,WAITTM,0
12803  062236  033544  033550  033546
12804  062244  001126  033552  000000
12805  062252  001116  004504  004500  DT5:    .WORD   $ERRPC,TSTNM,REGADR,$GDDAT,$BDDAT,0
12806  062260  001124  001126  000000
12807  062266  001116  004504  004500  DT6:    .WORD   $ERRPC,TSTNM,REGADR,$BDDAT,0
12808  062274  001126  000000
12809  062300  001116  004504  001200  DT7:    .WORD   $ERRPC,TSTNM,$TMP1,0
12810  062306  000000
12811  062310  001116  004504  001122  DT10:   .WORD   $ERRPC,TSTNM,$BDADR,CS1,CS2,DS1,ER1,0
12812  062316  002262  002260  002304
12813  062324  002264  000000
12814  062330  001116  004504  002262  DT26:   .WORD   $ERRPC,TSTNM,CS1,CS2,DS1,ER1,ER2,ER3,0
12815  062336  002260  002304  002264
12816  062344  002270  002276  000000
12817  062352  001116  004504  004502  DT30:   .WORD   $ERRPC,TSTNM,ERWORD,$GDDAT,$BDDAT,0
```

```
12818  062360  001124  001126  000000
12819  062366  001116  004504  004500  DT51:   .WORD   $ERRPC,TSTNM,REGADR,$GDDAT,$BDDAT,ILLEGL,0
12820  062374  001124  001126  002364
12821  062402  000000
12822  062404  001116  004504  004500  DT60:   .WORD   $ERRPC,TSTNM,REGADR,$GDDAT,$BDDAT,$BDADR,0
12823  062412  001124  001126  001122
12824  062420  000000
12825  062422  001116  004504  033350  DT61:   .WORD   $ERRPC,TSTNM,PCJSR,$BDADR,0
12826  062430  001122  000000
12827  062434  001116  004504  033350  DT62:   .WORD   $ERRPC,TSTNM,PCJSR,$BDADR,0
12828  062442  001122  000000
12829  062446  001116  004504  002262  DT65:   .WORD   $ERRPC,TSTNM,CS1,AS,DS1,0
12830  062454  002300  002304  000000
12831  062462  001116  004504  002266  DT66:   .WORD   $ERRPC,TSTNM,DST,ER1,ER2,ER3,CS1,CS2,0
12832  062470  002264  002270  002276
12833  062476  002262  002260  000000
12834  062504  001116  004504  002262  DT72:   .WORD   $ERRPC,TSTNM,CS1,CS2,DS1,DST,CA,ER1,WC,0
12835  062512  002260  002304  002266
12836  062520  002274  002264  002254
12837  062526  000000
12838
12839  062530      000      000      000  DF1:    .BYTE   0,0,0,0,0,0,0,0
12840  062533      000      000      000
12841  062536      000
12842  062537      000      000      000  DF4:    .BYTE   0,0,0,0,0,1,0
12843  062542      000      000      001
12844  062545      000
12845  062546      000      000      000  DF5:    .BYTE   0,0,0,0,0
12846  062551      000      000
12847  062553      000      000      000  DF6:    .BYTE   0,0,0,0
12848  062556      000
12849  062557      000      000      000  DF7:    .BYTE   0,0,0
12850  062562      000      000      000  DF10:   .BYTE   0,0,0,0,0,0,0,0
12851  062565      000      000      000
12852  062570      000
12853
12854  062571      000      000      000  DF26:   .BYTE   0,0,0,0,0,0,0,0,0
12855  062574      000      000      000
12856  062577      000      000
12857
12858  062601      000      000      000  DF30:   .BYTE   0,0,0,0,0
12859  062604      000      000
12860
12861  062606      000      000      000  DF51:   .BYTE   0,0,0,0,0,0
12862  062611      000      000      000
12863
12864  062614      000      000      000  DF60:   .BYTE   0,0,0,0,0,0
12865  062617      000      000      000
12866  062622      000      000      000  DF61:   .BYTE   0,0,0,0
12867  062625      000
12868  062626      000      000      000  DF62:   .BYTE   0,0,0,0
12869  062631      000
12870  062632      000      000      000  DF65:   .BYTE   0,0,0,0,0
12871  062635      000      000
12872  062637      000      000      000  DF66:   .BYTE   0,0,0,0,0,0,0,0,0
12873  062642      000      000      000
```

```
 12874  062645      000      000      000
 12875
 12876  062650      000      000      000  DF72:    .BYTE   0,0,0,0,0,0,0,0,0
 12877  062653      000      000      000
 12878  062656      000      000      000
 12879
 12880            062662                         .EVEN
 12881
 12882            000001                    .END
```

```
ACL    = 000040            1824#
ACU    = 100000            1784#
ADDMOD   000204             798#   2214
ADTIMO   036450           10363   10419#
AOE    = 001000            1701#   6187    6455
AS       002300            1903#  10145*  11082   12829
ATA    = 100000            1689#   4111    4450    4594    4804    5368    5512    5697    5836    6195    6447    6902    7193
                           7510    7807    8318    8488    9822
ATABLE   004566            1972#   2521
ATTENT   004644            2006#   2466*   2521*   4136    4434    4610    4806    5379    5525    5710    5847    6201    6481
                           6911    7209    7514    7811    8322    8504
AT0    = 000001            1721#
AT1    = 000002            1722#
AT2    = 000004            1723#
AT3    = 000010            1724#
AT4    = 000020            1725#
AT5    = 000040            1726#
AT6    = 000100            1727#
AT7    = 000200            1728#
A16    = 000400            1664#
A17    = 001000            1665#
BA       002256            1892#  11010
BAI    = 000010            1630#   8830
BASECH   035460             798   10348#  10425
BEGIN    004712             797    2028#  10418   11363
BEGIN1   004662             801    2022#
BEGIN2   004676             799    2025#
BIT0   = 000001             757#   3021    3177    7491    7793
BIT00  = 000001             747#    757
BIT01  = 000002             746#    756
BIT02  = 000004             745#    755
BIT03  = 000010             744#    754
BIT04  = 000020             743#    753
BIT05  = 000040             742#    752
BIT06  = 000100             741#    751
BIT07  = 000200             740#    750
BIT08  = 000400             739#    749   10472
BIT09  = 001000             738#    748    2385   10480   10963
BIT1   = 000002             756#   7469    7771
BIT10  = 002000             737#   7469    7771   10947
BIT11  = 004000             736#  10487
BIT12  = 010000             735#
BIT13  = 020000             734#  10954
BIT14  = 040000             733#  10458
BIT15  = 100000             732#
BIT2   = 000004             755#   7469
BIT3   = 000010             754#
BIT4   = 000020             753#   9906
BIT5   = 000040             752#
BIT6   = 000100             751#
BIT7   = 000200             750#
BIT8   = 000400             749#
BIT9   = 001000             748#
BPTVEC= 000014              764#
CA       002274            1901#  11074   12834
CC       002316            1910#  11114
```

```
CHECK   033352          2742    9803#
CHECKC  033410          9807    9813#
CHECKT  033374          2970    3125    3249    3353    3500    3646    3708    3966    4075    4245    4347    4541    4684
                        4737    4939    5088    5318    5462    5574    5648    5791    5950    6127    6393    6594    6752
                        6824    6997    7148    7334    7365    7448    7678    7750    7909    8075    8289    8423    8601
                        8735    8837    8928    9003    9126    9232    9329    9462    9809#
CHREG   034204          2809    3009    3016    3165    3172    4106    4115    4124    4388    4410    4420    4438    4445
                        4582    4589    4603    4792    4799    5363    5372    5507    5518    5692    5703    5821    5831
                        5840    6167    6176    6190    6433    6442    6881    6890    6897    6915    7179    7188    7202
                        7486    7493    7505    7518    7788    7795    7802    7815    8306    8313    8329    8340    8474
                        8483    8497    8508    10064#
CKSWR = 104407          10457   10938   10962   11321#
CLAREA  033164          2906    3067    3294    3312    3320    3557    3803    3932    4043    4210    4315    4494    4500
                        4508    4886    4906    5055    5251    5430    5615    5759    5922    6021    6064    6308    6316
                        6324    6548    6556    6727    6791    6957    6965    7108    7116    7310    7402    7408    7416
                        7635    7653    7711    7717    7871    7877    8038    8045    8977    9079    9100    9303    9418
                        9436    9702#
CLDISK  033314          2240    2257    2462    2602    2644    2689    2738    2867    3055    3244    3277    3458    3640
                        3677    3913    4038    4191    4477    4678    4715    4869    5050    5214    5423    5567    5608
                        5752    5905    5978    6271    6331    6527    6563    6709    6781    6951    7101    7283    7295
                        7605    7617    7864    8031    8203    8238    8401    8543    8562    8649    8696    8786    8813
                        8914    8957    9068    9228    9284    9388    9782#
CLR   = 000040          1632#   2661    2673    2705    2713    9787
COMPAR  035342          3207    3608    3860    4162    4640    5035    5184    6510    6672    7073    7241    7562    8001
                        8176    9177    9501    10310#
COMREG  034312          2821    3030    3186    3428    3589    3840    4015    4142    4293    4456    4617    4812    5014
                        5163    5391    5532    5717    5854    6243    6489    6641    6930    7051    7221    7538    7840
                        7984    8159    8363    8527    10132#
CPHALT  057465          2743    2972    3127    3251    3355    3502    3648    3710    3968    4077    4247    4349    4543
                        4686    4739    4941    5090    5320    5464    5576    5650    5793    5952    6129    6395    6596
                        6754    6826    6999    7150    7336    7367    7450    7680    7752    7911    8077    8291    8425
                        8603    8737    8839    8930    9005    9128    9234    9331    9464    12539#
CR    = 000015           672#   10632   10642
CRLF  = 000200           673#   10603   10642
CSF   = 000002          1771#
CSU   = 000010          1773#
CS1     002262          1896#   8636*   8637    8770*   8771    8880*   8881    9046    9162    11026   12799   12811   12814
                        12829   12831   12834
CS2     002260          1893#   11018   12811   12814   12831   12834
DB      002252          1890#
DCK   = 100000          1707#
DCL   = 000100          1825#
DCLEAR  002330          1925#
DDISP = 177570           679#     876    2060
DE1   = 000040          1679#
DFF20 = 000002          1675#
DFF5  = 000001          1674#
DF1     062530           934     946     957    12839#
DF10    062562          1011    1025    1039    1055   12850#
DF26    062571          1174   12854#
DF30    062601          1195    1214    1230    1247    1308    1342    1366    1402    1438    1545    1556   12858#
DF4     062537           970   12842#
DF5     062546           980    1065    1077    1091    1105    1117    1135    1132    1142    1151    1162    1185    1206
                        1224    1240    1258    1269    1283    1295    1320    1333    1355    1395    1414    1426    1451
                        1492    1504    1570   12845#
DF51    062606          1381   12861#
```

```
DF6     062553            990   12847#
DF60    062614           1468   12864#
DF61    062622           1476   12866#
DF62    062626           1484   12868#
DF65    062632           1516   12870#
DF66    062637           1534   12872#
DF7     062557            998   12849#
DF72    062650           1586    1597    1604   12876#
DH1     057737            927     939     951   12575#
DH10    060536           1003    1017    1031    1047  12644#
DH26    060715           1166   12665#
DH30    061115           1190    1212    1228    1245  1306  1340  1364  1400  1436  1543  1554  12688#
DH4     060117            964   12595#
DH5     060260            975    1060    1072    1086  1100  1112  1123  1130  1140  1149  1160  1183  1204
                         1222    1238    1256    1267  1281  1293  1318  1331  1353  1393  1412  1424  1449
                         1490    1502    1565   12612#
DH51    061227           1375   12703#
DH6     060377            986   12626#
DH60    061366           1462   12721#
DH61    061523           1472   12738#
DH62    061615           1480   12749#
DH65    061670           1510   12757#
DH66    062004           1526   12771#
DH7     060476            995   12737#
DH72    062102           1577    1585    1602   12783#
DIGB  = 000004           1676#
DISPLA  001142            876#   2060*   2068*  10501*  10946*
DISPRE  000174            780#   2068
DLT   = 100000           1642#
DL64  = 000020           1678#
DMD   = 000001           1711#   8447    8513
DPR   = 000400           1682#   2798    3397    3722    4779  4982  5131  7952  8123  9823
DRY   = 000200           1681#   2798    3515    3670    4709  4761  5602  7393  8951  9263  9411  9823
DST     002266           1898#  11066   12831   12834
DSWR  = 177570            678#    875    2059
DS1     002304           1905#   9044    9160   11034   12811  12814  12829  12834
DT      002306           1906#   2585*  11138
DTE   = 010000           1704#
DTSY  = 001000           1717#
DT1     062210            933     945     956   12799#
DT10    062310           1010    1024    1038    1054  12811#
DT26    062330           1173   12814#
DT30    062352           1194    1213    1229    1246  1307  1341  1365  1401  1437  1544  1555  12817#
DT4     062230            969   12802#
DT5     062252            979    1064    1076    1090  1104  1116  1124  1131  1141  1150  1161  1184  1205
                         1223    1239    1257    1268  1282  1294  1319  1332  1354  1394  1413  1425  1450
                         1491    1503    1569   12805#
DT51    062366           1380   12819#
DT6     062266            989   12807#
DT60    062404           1467   12822#
DT61    062422           1475   12825#
DT62    062434           1483   12827#
DT65    062446           1515   12829#
DT66    062462           1533   12831#
DT7     062300            997   12809#
DT72    062504           1585    1596    1603   12834#
```

```
DVA    = 004000          1667#   2667    2789    3388    3721    4770    4973    5122    7943    8114    9815
ECH    = 000100          1698#
EC1    = 004000          1797#   2933    2953    3092    3111    3339    3472    3473    3474    3691    3692    3693    3951
                         4061    4229    4333    4525    4526    4527    4925    5073    5074    5075    5281    5300    5448
                         5633    5634    5635    5776    5777    5778    5944    6093    6112    6356    6357    6358    6375
                         6376    6377    6580    6581    6582    6746    6809    6982    6983    6984    7133    7134    7135
                         7328    7432    7433    7434    7672    7735    7894    7895    7896    8061    8062    8063    8595
                         8729    8831    8832    8833    8997    9119    9120    9121    9323    9455    9456    9457
EC1      002312          1908#   11122
EC2      002314          1909#   11130
EMTVEC=  000030           767#    2044*   2045*   2192*   2234*
EM1      043014           925    11382#
EM10     043534          1001    11441#
EM11     043603          1014    11449#
EM12     043710          1028    11461#
EM13     044016          1042    11473#
EM14     044135          1058    11487#
EM15     044207          1068    11495#
EM16     044350          1080    11513#
EM17     044476          1094    11529#
EM2      043043           937    11386#
EM20     044633          1108    11546#
EM21     045002          1121    11565#
EM22     045055          1128    11573#
EM23     045130          1135    11581#
EM24     045277          1145    11599#
EM25     045442          1154    11617#
EM26     045624          1165    11637#
EM27     045714          1177    11647#
EM3      043132           949    11396#
EM30     046132          1188    11673#
EM31     046212          1198    11682#
EM32     046427          1209    11707#
EM33     046534          1217    11719#
EM34     046736          1227    11743#
EM35     046762          1233    11747#
EM36     047200          1243    11772#
EM37     047255          1250    11780#
EM4      043213           960    11405#
EM40     047472          1261    11805#
EM41     047701          1272    11830#
EM42     050162          1286    11862#
EM43     050440          1298    11892#
EM44     050776          1311    11931#
EM45     051270          1323    11964#
EM46     051576          1336    11999#
EM47     051722          1345    12014#
EM5      043324           973    11418#
EM50     052204          1358    12045#
EM51     052423          1369    12072#
EM52     052670          1385    12101#
EM53     053141          1398    12130#
EM54     053227          1405    12140#
EM55     053474          1417    12169#
EM56     053762          1429    12201#
EM57     054172          1441    12225#
```

```
EM6      043376            983   11425#
EM60     054464           1454   12258#
EM61     055073           1471    1479   12306#
EM63     055157           1488   12315#
EM64     055232           1496   12323#
EM65     055451           1509   12348#
EM66     055473           1520   12352#
EM67     055704           1536   12377#
EM7      043464            993   11434#
EM70     056160           1547   12409#
EM71     056366           1559   12434#
EM72     056627           1573   12463#
EM73     057075           1593   12492#
EM74     057245           1600   12511#
ERFLG$   004632           1993#   8429*  10941*
ERR    = 040000           1688#   4113    4452    4596    5370    5514    5699    5838    6197    6449    6904    7195    7512
                          7809    8320    8490    9044    9160
ERRVEC= 000004            760#   2057    2058*   2069*   2194*   2202*   2236*  10463   10464*  10466*  10469*
ERWORD   004502           1954#  10323*  10329*  12817
ER1      002264           1897#   8870*   8871   11042   12811   12814   12831   12834
ER2      002270           1899#  11050   12814   12831
ER3      002276           1902#  11058   12814   12831
EXT1   = 000001           1751#
EXT10  = 000010           1754#
EXT2   = 000002           1752#
EXT20  = 000020           1755#
EXT4   = 000004           1753#
EXT40  = 000040           1756#
FEN    = 000200           1777#
FER    = 000020           1696#   6920    7207
FILL     034250          10094#
FILLRE   033216           2999    3004    3023    3155    3160    3179    3410    3415    3420    3565    3570    3575    3580
                          3811    3816    3821    3826    3831    3997    4002    4007    4131    4276    4281    4286    4380
                          4384    4400    4405    4429    4572    4577    4598    4995    5000    5005    5144    5149    5154
                          6157    6162    6185    6214    6219    6226    6231    6423    6428    6453    6466    6473    6624
                          6629    6634    6856    6860    6869    6874    6906    7027    7032    7037    7197    7500    7525
                          7530    7822    7827    7832    7965    7971    7976    8136    8141    8146    8151    8350    8356
                          8492    9730#
FINACC   004564           1966#   9881*
FINALA   004562           1965#   9882*
FIRST    004634           1995#   2081   2118*
FLHEAD   033140           2881    2893    3062    3284    3302    3549    3795    3920    4198    4484    4876    4898    5230
                          5241    5912    5998    6010    6042    6054    6284    6296    6540    6717    7300    7625    7643
                          8573    8707    8966    9089    9292    9427   9673#
FMT22  = 010000           1798#   2932    2933    2934    2952    2953    2954    3091    3092    3093    3110    3111    3112
                          3338    3339    3340    3472    3473    3474    3691    3692    3693    3950    3951    3952    4060
                          4061    4062    4228    4229    4230    4332    4333    4334    4525    4526    4527    4924    4925
                          4926    5073    5074    5075    5280    5281    5282    5299    5300    5301    5447    5448    5449
                          5633    5634    5635    5776    5943    5944    5945    6092    6093    6094    6111    6112    6113
                          6356    6357    6358    6375    6376    6377    6580    6581    6582    6745    6746    6747    6808
                          6982    6983    6984    7133    7327    7328    7329    7432    7433    7434    7671    7672    7673
                          7734    7735    7736    7894    7895    7896    8061    8062    8063    8411    8430    8594    8595
                          8596    8728    8729    8730    8831    8832    8833    8996    8997    8998    9119    9120    9121
                          9322    9323    9324    9455    9456    9457    9536
FUTABL   002322           1921#   8216
GNS    = ****** U          779    2089    2093    2098    2102    2106    2111    2115    2168    2211    2226    2232    2308
```

```
                    2334    2338    2342    2346    2399    2405    2411    2490    2494    2524    2530    2536    2563
                    2569    2575    2609    2613    2617    2623    9578    9584   10199   10205   10212   10216   10220
                   10227   10231   10351   10357   10374   10380   10386   10392   10398   10402   10406   10410   10416
                   10422   10431   11000   11008   11016   11024   11032   11040   11048   11056   11064   11072   11080
                   11088   11096   11104   11112   11120   11128   11136   11144   11313   11314   11315   11316   11317
                   11319   11321   11322   11323   11324   11325
GO      = 000001    1661#   2768    2982    3137    3263    3367    3388    3492    3506    3662    3721    3727    3756
                    3980    4089    4259    4361    4555    4698    4751    4770    4953    4973    5102    5122    5332
                    5476    5593    5661    5805    5964    6141    6407    6608    6766    6838    7011    7162    7348
                    7384    7462    7692    7764    7923    7943    8090    8114    8294    8452    8615    8749    8849
                    8942    9017    9140    9254    9343    9402    9481    9906
GRV     = 000010    1677#
GTSWR   = 104406    2125   11319#
HCE     = 000200    1699#   4129    4443    4587
HCI     = 002000    1796#   2934    2954    3093    3112    3340    3474    3693    3952    4062    4230    4334    4527
                    4926    5075    5282    5301    5449    5635    5778    5945    6094    6113    6358    6377    6582
                    6747    6810    6984    7135    7329    7434    7673    7736    7896    8063    8596    8730    8833
                    8998    9121    9324    9457
HCRC    = 000400    1700#
HT      = 000011     670#   10601   10642
HT.     = 015440    2778#   2783    2990#   2995    3145#   3150    3269#   3274    3377#   3382    3513#   3518    3540#
                    3545    3668#   3673    3761#   3766    3786#   3791    3988#   3993    4097#   4102    4267#   4272
                    4369#   4374    4563#   4568    4707#   4712    4759#   4764    4962#   4967    5111#   5116    5339#
                    5344    5483#   5488    5600#   5605    5668#   5673    5812#   5817    5971#   5976    6148#   6153
                    6414#   6419    6615#   6620    6773#   6778    6845#   6850    7018#   7023    7169#   7174    7356#
                    7361    7391#   7396    7477#   7482    7700#   7705    7779#   7784    7932#   7937    8103#   8108
                    8297#   8302    8625#   8630    8759#   8764    8860#   8865    8949#   8954    9031#   9036    9147#
                    9152    9261#   9266    9355#   9360    9409#   9414    9488#   9493
IAE     = 002000    1702#   5341    5377    5485    5523    5670    5708    5814    5845
IE      = 000100    1662#   2654    2699    2982    3137    3263    3367    3388    3492    3506    3662    3721    3727
                    3756    3980    4089    4259    4361    4555    4698    4751    4770    4953    4973    5102    5122
                    5332    5476    5593    5661    5805    5964    6141    6407    6608    6766    6838    7011    7162
                    7348    7384    7462    7692    7764    7923    7943    8090    8114    8294    8615    8749    8849
                    8942    9017    9140    9254    9343    9402    9481    9933    9945   10018   10041
ILF     = 000001    1692#   8311
ILLEGL  002364      1939#   8228*   8244    8255    8263    8293    8335   12819
ILR     = 000002    1693#
IOTVEC= 000020       765#   2042*   2043*
IR      = 000100    1633#   2244    4395    6183    6922    6923    7042    7043    7213    7214    8261    8272    8347
                    8520    8521
IXE     = 004000    1781#
JP1       015624    4378    4398#
JP2       015660    4396    4418#
JP3       023352    6854    6867#
JP4       023372    6863    6879#
KIPAR0= 172340       837#
KIPAR1= 172342       838#
KIPAR2= 172344       839#
KIPAR3= 172346       840#
KIPAR4= 172350       841#
KIPAR5= 172352       842#
KIPAR6= 172354       843#
KIPAR7= 172356       844#
KIPDR0= 172300       826#
KIPDR1= 172302       827#
KIPDR2= 172304       828#
```

```
KIPDR3= 172306          829#
KIPDR4= 172310          830#
KIPDR5= 172312          831#
KIPDR6= 172314          832#
KIPDR7= 172316          833#
LA      002320         1911#  11106
LBT   = 002000         1684#   2992   3014   3170   6199   6451   9822
LF    = 000012          671#  10636  10642
LT.   = 000000         2778#   2783   2990#  2995   3145#  3150   3269#  3274   3377#  3382   3513#  3518   3540#
                       3545    3668#  3673   3761#  3766   3786#  3791   3988#  3993   4097#  4102   4267#  4272
                       4369#   4374   4563#  4568   4707#  4712   4759#  4764   4962#  4967   5111#  5116   5339#
                       5344    5483#  5488   5600#  5605   5668#  5673   5812#  5817   5971#  5976   6148#  6153
                       6414#   6419   6615#  6620   6773#  6778   6845#  6850   7018#  7023   7169#  7174   7356#
                       7361    7391#  7396   7477#  7482   7700#  7705   7779#  7784   7932#  7937   8103#  8108
                       8297#   8302   8625#  8630   8759#  8764   8860#  8865   8949#  8954   9031#  9036   9147#
                       9152    9261#  9266   9355#  9360   9409#  9414   9488#  9493
MCLK  = 000002         1712#
MCPE  = 020000         1668#
MHS   = 001000         1779#
MINX  = 000004         1713#   8448   8449   8460   8461   8464   8465   8468   8469
MMVEC = 000250          815#
MOL   = 010000         1686#   2603   2619   2798   3397   3722   4779   4982   5131   7952   8123   9823
MPE   = 000400         1635#
MR      002302         1904#  11098
MRD   = 000020         1715#
MSE   = 000020         1774#
MSTCK = 000010         1714#
MWR   = 000040         1716#
MXF   = 001000         1636#   8260   8270   8279   8345
NED   = 010000         1639#
NEM   = 004000         1638#
NHS   = 002000         1780#
NOPERA  002322         1922#
NOPUSH  004624         1986#   2022*  2026*  2029*
NOUNIT  004620         1982#   2362*  2419*  2425   2431   2500*  9602*
NOUSE   057420         2391   12531#
NUNIT   004622         1984#   2431*  2432*
OCYL  = 100000         1827#
OF      002272         1900#  11090
OFREV = 000200         1795#
OFSET   033300         9777#
OFSETC  002354         1935#   9778
OFSTVL  004510         1961#
OF100 = 000004         1790#
OF200 = 000010         1791#
OF25  = 000001         1788#
OF400 = 000020         1792#
OF50  = 000002         1789#
OF800 = 000040         1793#
OPERSE  034450        10194#  10684  10767
OPI   = 020000         1705#   8502   8871
OR    = 000200         1634#   4393   4415   6181   6895   6922   6923   7042   7043   7213   7214   8260   8270
                       8520    8521
PAR   = 000010         1695#
PCJSR   033350         9801#   9803*  9804*  9809*  9810*  12825  12827
PCLBUF  002246         1880#   9905*
```

```
PCLCSR  002244          1879#   9879*   9906*
PCLCTR  002250          1881#   9880
PGE   = 002000          1637#
PIP   = 020000          1687#
PIRQ  = 177772           677#
PIRQVE= 000240           771#
PKACK   002360          1937#   2746    2767    2788
PLU   = 020000          1783#
PRE   = 000020          1823#
PRITEM  041170          2160*   9601*  10196*  10987#  11149*  11167   11171*
PROG  = 001000          1683#
PR0   = 000000           694#
PR1   = 000040           695#
PR2   = 000100           696#
PR3   = 000140           697#
PR4   = 000200           698#
PR5   = 000240           699#
PR6   = 000300           700#
PR7   = 000340           701#
PS    = 177776           674#    675    2073*   2653*   2698*   2733*   3520*   3769*   9575*  10195*
PSEL  = 002000          1666#
PSU   = 000001          1820#
PSW   = 177776           675#
PUTREG  035236          8639    8773    8873    8883    9042    9158   10144   10252#
PWRVEC= 000024           766#   2048*   2049*  11330*  11331*  11340*  11346*  11358*  11359*
RA      000200           797#  10412
RDCHR = 104410         10842   11322#
RDLIN = 104411         10889   11323#
RDOCT = 104412          2170   10222   10233   10360   10382   11324#
RDY   = 000200          1663#   2654    2667    2699    2789    3147    3271    3379    3542    3721    3763    3788    3990
                        4099    4269    4371    4565    4770    4964    5113    5973    6150    6416    6617    6775    6847
                        7020    7171    7358    7479    7702    7781    7934    8105    8299    8627    8761    8862    9033
                        9149    9357    9490    9815
READAT  002346          1932#   3094    3113    3136    4063    4088    5779    5804    6985    7010    7136    7161    7435
                        7461    7897    7922    7942    8064    8089    8113
READIN  002362          1938#
RECALI  002326          1924#   3262    4697    8941    9253    9401
REFOR   002350          1933#   3491    3755    4528    4554    5076    5101    5121    5636    5660    6359    6378    6406
                        6583    6607    8834    8848    9122    9139    9458    9480
REGADR  004500          1953#   2277*   2795*   2804*   3394*   3403*   3735*   3749*   4776*   4785*   4979*   4988*   5128*
                        5137*   7949*   7958*   8120*   8129*  10153*  12805   12807   12819   12822
REGSAV  041024         10940#
REGSA1  041032          2192   10942#
REINTO  003434          1947#   2907    3087    3106    3162    3209    3321    3468    3572    3610    3687    3818    3862
                        3933    4056    4164    4211    4509    4521    4579    4642    4899    4907    5036    5056    5069
                        5151    5186    5616    5629    5760    5772    6325    6352    6371    6425    6512    6557    6576
                        6631    6674    6966    6978    7029    7075    7117    7129    7243    7417    7428    7527    7564
                        7890    7973    8003    8046    8057    8143    8178    8827    9080    9115    9179    9419    9451
                        9503
RELEAS  002332          1926#
RESVEC= 000010           761#
RETCL   002356          1936#
RHAS    002216          1861#   2269    2310    8098*
RHBA    002174          1849#   3005    3161    3416    3571    3817    4003    4282    4385    4406    4578    5001    5150
                        5356    5500    5685    6158    6424    6630    6861    6875    7028    7526    7828    7972    8142
                        8243*   8266*   8275*   8351    8409*   8518   10282*
```

```
RHBAE   002240          1873#  2222
RHCA    002212          1859#  3017   3173   6215   6227   6467   6474   8431*  9750*  9768*  10279*
RHCC    002234          1868#  3832   6220   6232   7487   7789   8152   9881   10258
RHCS1   002200          1854#  2746*  2770*  2984*  3139*  3146   3265*  3369*  3378   3508*  3533*  3541   3576
                        3664*  3729*  3787   3822   3982*  3989   4091*  4098   4116   4261*  4268   4363*  4370
                        4421   4557*  4564   4604   4700*  4753*  4793   4955*  4963   5104*  5112   5334*  5360
                        5478*  5504   5595*  5663*  5689   5807*  5822   5966*  5972   6143*  6149   6168   6409*
                        6415   6434   6610*  6616   6768*  6774   6840*  6846   6882   7013*  7019   7164*  7170
                        7180   7287   7350*  7357   7386*  7464*  7478   7494   7610   7694*  7701   7766*  7780
                        7796   7925*  7933   8092*  8104   8298   8330   8452*  8475   8617*  8626   8636   8751*
                        8760   8770   8851*  8861   8880   8944*  9019*  9032   9142*  9148   9256*  9345*  9356
                        9404*  9483*  9489   9752*  9769*  9778*  9782   9933   9945   10018  10041  10288* 10353
                        10364  10367  10388
RHCS2   002176          1850#  2244   2311   2368*  2509*  4389   4411   5358   5502   5687   6177   6891   6921
                        7041   7212   8341   8519   9783   10285*
RHCS3   002242          1874#
RHDB    002170          1847#  2197   2242*  10361
RHDST   002204          1856#  3024   3180   3421   3524*  3581   3770*  3827   4008   4132   4287   4430   4599
                        5006   5155   5382   5515   5700   5829   6239   6484   6635   6907   7038   7198   7501
                        7823   7977   8147   8422*  8493   9751*  10280*
RHDS1   002222          1863#  2385   2779   2810   2991   3010   3166   3270   3514   3669   3762   4107   4446
                        4590   4708   4760   4800   5364   5508   5601   5693   5832   6191   6443   6898   7189
                        7392   7506   7803   8314   8484   8950   9262   9410   9784
RHDT    002224          1864#  2369   2371   2375   2377   2380   2382   2407   2510   2512   2538   2545   2547
                        2550   2552   2555   2557   2583   2585
RHEC1   002230          1866#
RHEC2   002232          1867#
RHER1   002202          1855#  2314   4125   4439   4583   5340   5373   5484   5519   5669   5704   5813   5841
                        6186   6454   6916   7203   7519   7816   8307   8498   8870   9785
RHER2   002206          1857#
RHER3   002214          1860#
RHLA    002236          1869#  9882
RHMR    002220          1862#  8446   8509
RHOF    002210          1858#  8430*  9777*  10286*
RHSN    002226          1865#  2532   2582   2584
RHWC    002172          1848#  2753   2964   3000   3120   3156   3347   3411   3482   3566   3701   3812   3960
                        3998   4069   4239   4277   4341   4381   4401   4535   4573   4730   4933   4996   5082
                        5145   5312   5354   5456   5498   5642   5683   5785   6121   6163   6387   6429   6589
                        6625   6818   6857   6870   6991   7033   7142   7442   7531   7744   7833   7903   7966
                        8069   8137   8242*  8265*  8274*  8283   8357   8407*  8435   8517   9735   10069  10153
                        10256  10258  10281*
RH70    004640          2000#  2132*  2221*  2223*  4377   6853   8208
RH70CK  006056          2130#
RMR   = 000004          1694#  7523   7820
RPTCTR  034014          9990#  10036
RPTRP1  011514          2650   2666#
RPTRP2  011622          2695   2710#
RPVEC   002166          1611#  2075   2649   2694   2759*  2974*  3129*  3254*  3358*  3499*  3526*  3650*  3712*
                        3773*  3971*  4080*  4250*  4352*  4546*  4689*  4742*  4944*  5093*  5323*  5467*  5579*
                        5653*  5796*  5955*  6132*  6398*  6599*  6757*  6829*  7002*  7153*  7339*  7375*  7453*
                        7683*  7755*  7914*  8080*  8606*  8740*  8841*  8933*  9008*  9131*  9245*  9334*  9393*
                        9467*  10376  10383*  10394  10434*
RPVECT  036534          2076   10428#  10434
RP06    004636          1998#  2508*  2515*  2875   2917   3077   5224   5263   5992   6036   6076   6208   6278
                        6341   6461   9272   9522   9542
RP4VEC  004506          1956#  2159*  2759   2974   3129   3254   3358   3526   3650   3773   3971   4080   4250
```

```
                    4352    4546    4689    4742    4944    5093    5323    5467    5579    5653    5796    5955    6132
                    6398    6599    6757    6829    7002    7153    7339    7375    7453    7683    7755    7914    8080
                    8606    8740    8841    8933    9008    9131    9245    9334    9393    9467
RTN       006546    2228    2234#
RUN       035276    2922    2942    3082    3101    3328    3463    3682    3940    4051    4218    4323    4515    4914
                    5063    5270    5289    5438    5623    5767    5933    6082    6101    6346    6365    6570    6735
                    6799    6973    7124    7317    7423    7661    7725    7885    8052    8584    8718    8821    8986
                    9109    9312    9445   10279#
RUNCTR    004642    2003#   8453*   8456*
RUNWAT    027642    8453#
SAVER     033462    2752    2963    3119    3346    3481    3700    3959    4068    4238    4340    4534    4729    4932
                    5081    5311    5455    5641    5784    6120    6386    6588    6817    6990    7141    7441    7743
                    7902    8068    8282    8434    9844#
SAVERE    004512    1964#   2754    2823    2965    3032    3121    3188    3348    3430    3483    3591    3702    3842
                    3961    4017    4070    4136*   4144    4240    4295    4342    4434*   4458    4536    4610*   4619
                    4731    4806*   4814    4934    5016    5083    5165    5313    5354*   5356*   5358*   5360*   5379*
                    5382*   5393    5457    5498*   5500*   5502*   5504*   5515*   5525*   5534    5643    5683*   5685*
                    5687*   5689*   5700*   5710*   5719    5786    5829*   5847*   5856    6122    6201*   6239*   6245
                    6388    6481*   6484*   6491    6590    6643    6819    6911*   6923*   6924*   6932    6992    7043*
                    7044*   7053    7143    7209*   7214*   7215*   7223    7443    7514*   7540    7745    7811*   7842
                    7904    7986    8070    8161    8284    8322*   8335*   8365    8436    8504*   8517*   8518*   8521*
                    8522*   8529    9736*  10072*  10074*  10145
SC      = 100000    1670#   4120    4425    4608    4797    5826    6172    6438    6886    7184    7498    7800    8247
                    8257    8267    8276    8334    8479    8637    8771    8881
SC1     = 000100    1757#
SC10    = 001000    1760#
SC2     = 000200    1758#
SC20    = 002000    1761#
SC4     = 000400    1759#
SEECOM    002352    1934#   3661    5592    7383    9769
SEEKCY    033264    3657    5587    7371    9768#
SELECT    004626    1988#   2023*   2025*   2028*   2162    2435    2473    9590
SELTST    010126    2328    2435#
SERCH     002334    1927#   3475    3505    3694    3719    3726    4750    4769    9752
SILOSZ    004650    2008#   2241*   2242    2243*   2247    2251    2252    2254
SKI     = 040000    1826#
SN        002310    1907#   2584*  11146
SND1      006020    2084    2118#
SRCH      033244    3714    4722    9750#
SR0     = 177572     819#
SR1     = 177574     820#
SR2     = 177576     821#
SR3     = 172516     822#
SST1      031354    9074#
SST2      031362    9077#   9084    9207
SST3      031402    9063*   9087#   9198*
SST4      031422    9064*   9098#   9199*
SST5      031434    9065*   9107#   9200*
STACK   = 001000     665#   2040    2191    2267    2459    2642    2687    2735    2864    3241    3637    3910    4188
                    4675    4866    5211    5420    5564    5749    5902    6706    7098    7280    7602    8028    8200
                    8398    8559    8693    8911    9225
START     004722    2024    2027    2032#
STKLMT= 177774     676#
ST22      024376    7293#   7576
ST22A     022774    6639#
ST23      025154    7566    7573#
```

```
ST24     025226            7615#   8016
ST28     026242            8006    8013#
SWR      001140             875#   2038    2059*   2061    2067*   2123    2130    2206    2304    2326    10157   10332   10458
                           10472   10474   10480   10487   10687   10726   10781*  10947   10954   10959   10963   10989   11338
                           11351*
SWREG    000176             781#   2067    2123    10687   10726   10749
SW0    = 000001             729#
SW00   = 000001             719#    729
SW01   = 000002             718#    728
SW02   = 000004             717#    727
SW03   = 000010             716#    726
SW04   = 000020             715#    725
SW05   = 000040             714#    724
SW06   = 000100             713#    723    10991
SW07   = 000200             712#    722    10159   10334
SW08   = 000400             711#    721
SW09   = 001000             710#    720
SW1    = 000002             728#
SW10   = 002000             709#
SW11   = 004000             708#
SW12   = 010000             707#   2130
SW13   = 020000             706#   2206    2304    2326
SW14   = 040000             705#
SW15   = 100000             704#
SW2    = 000004             727#
SW3    = 000010             726#
SW4    = 000020             725#
SW5    = 000040             724#
SW6    = 000100             723#
SW7    = 000200             722#
SW8    = 000400             721#
SW9    = 001000             720#
TBITVE= 000014              762#
TDF    = 000040            1775#
TESTAD   034446           10193#
TIMCNT   034012            2656    2701    9922    9927    9989#   10007   10012
TIME1    033514            9879#
TIME2    033526            2159    9881#
TKVEC  = 000060             769#   10664*  10665*
TMP0     004652            2010#   3493*   3533    3755*   3756*   3779    7287*   7468    7471*   7549    7610*   7770    7773*
                           7851
TMP1     004654            2011#   8568*   8666*   8702*   8803*
TMP4     004656            2012#
TMP5     004660            2013#   3494*   3524    3757*   3770    7288*   7573*   7611*   8013*
TN     = 000037            2459#   2460    2735#   2736    2864#   2865    3241#   3242    3637#   3638    3910#   3911    4188#
                           4189    4675#   4676    4866#   4867    5211#   5212    5420#   5421    5564#   5565    5749#   5750
                           5902#   5903    6706#   6707    7098#   7099    7280#   7281    7602#   7603    8028#   8029    8200#
                           8201    8398#   8399    8559#   8560    8693#   8694    8911#   8912    9225#   9226
TOTALA   004646            2007#   2319*   2321*
TPVEC  = 000064             770#
TRAPVE= 000034              768#   2046*   2047*   2074*   3736*   3750*   3768*
TRE    = 040000            1669#   4122    4427    4608    5828    6174    6440    6888    7186    8257    8267    8276    8481
                           9046    9162
TRK1   = 004000            1762#
TRK10  = 040000            1765#
TRK2   = 010000            1763#
```

```
TRK20 = 100000          1756#
TRK4  = 020000          1764#
TRP     006510          2220    2229#
TRTVEC= 000014           763#
TSTNM   004504          1955#   2189*   2299*   2460*   2597*   2640*   2685*   2736*   2865*   3242*   3638*   3911*   4189*
                        4676*   4867*   5212*   5421*   5565*   5750*   5903*   6707*   7099*   7281*   7603*   8029*   8201*
                        8399*   8560*   8694*   8912*   9226*  10201   12799   12802   12805   12807   12809   12811   12814
                       12817   12819   12822   12825   12827   12829   12831   12834
TST1    006222          2164    2175    2186#   9653
TST10   011632          2707    2727#
TST11   012076          2863#
TST12   012752          3240#
TST13   013732          3636#
TST14   014500          3909#
TST15   015206          4187#
TST16   016336          4674#
TST17   016670          4865#
TST2    006714          2265#
TST20   017556          5210#
TST21   020136          5419#
TST22   020414          5563#
TST23   020752          5748#
TST24   021220          5901#
TST25   023040          6705#
TST26   024014          7097#
TST27   024342          7279#
TST3    006762          2273    2295#
TST30   025172          7575    7601#
TST31   026260          8015    8027#
TST32   026624          8199#
TST33   027452          8210    8397#
TST34   030100          8558#
TST35   030362          8692#
TST36   031020          8777    8910#
TST37   031632          9205    9224#
TST4    010142          2437    2455#   9599    9610
TST40   032564          9573#
TST5    011070          2595#
TST6    011422          2605    2637#
TST7    011532          2663    2669    2682#
TUF   = 000100          1776#
TYPDS = 104405          2390    2402    2418    2527    9581    9587    9597    9642   11193   11317#
TYPE  = 104401          2087    2091    2096    2100    2104    2109    2113    2166    2209    2224    2230    2306    2332
                        2336    2340    2344    2388    2391    2397    2403    2409    2416    2488    2492    2522    2528
                        2534    2561    2567    2573    2607    2611    2615    2621    2743    2972    3127    3251    3355
                        3502    3648    3710    3968    4077    4247    4349    4543    4686    4739    4941    5090    5320
                        5464    5576    5650    5793    5952    6129    6395    6596    6754    6826    6999    7150    7336
                        7367    7450    7680    7752    7911    8077    8291    8425    8603    8737    8839    8930    9005
                        9128    9234    9331    9464    9576    9582    9595    9598    9640    9643   10197   10203   10209
                       10210   10214   10218   10225   10229   10349   10355   10372   10378   10384   10390   10396   10400
                       10404   10408   10414   10420   10429   10563   10606   10681   10693   10747   10748   10751   10762
                       10772   10783   10802   10846   10849   10853   10918   10920   10949   10957   10998   11006   11014
                       11022   11030   11038   11046   11054   11062   11070   11078   11086   11094   11102   11110   11118
                       11126   11134   11142   11152   11174   11176   11179   11181   11196   11265   11313#  11360
TYPERR  042132          10994   11151#
TYPOC = 104402           2216    2408    2533    2539   10202   10208   10354   10377   10389   10395   10413   10433   10750
```

```
                    11003   11011   11019   11027   11035   11043   11051   11059   11067   11075   11083   11091   11099
                    11107   11115   11123   11131   11139   11147   11160   11189   11314#
TYPON = 104404      11316#
TYPOS = 104403      11315#
UNIB  = 000020       1631#
UNIT    004616       1981#  2172*  2177*  2430*  2439*  2468   2499*  2502   2526   9580   9604   9609*  9788
                    10283
UNITS   004576       1980#  2356   2360   2430   2480   9605
UNITSL  004630       1989#  2173*  2177   2439
UNLOAD  002324       1923#
UNS   = 040000       1706#
UPE   = 020000       1640#
US1   = 000001       1627#
US2   = 000002       1628#
US4   = 000004       1629#
UWR   = 000010       1822#
VAR1    015570       2246*  2247*  2248*  2249   4382#
VAR2    015600       2250*  2251*  2252*  2253   4386#
VAR3    023336       2249*  6858#
VAR4    023346       2253*  6862#
VAR5    015622       2256*  4396#
VUF   = 000002       1821#
VU30  = 010000       1782#
VV    = 000100       1680#  2780   2798   2814   3397   3722   4779   4982   5131   7952   8123   9806   9823
WAITBT  033550       9902#  9915*  9923   9928   10001* 10008  10013  12799  12802
WAITPC  033544       9900#  9912*  9913*  9998*  9999*  12799  12802
WAITRE  033546       9901#  9914*  9923   9928   9932   9944   10000* 10008  10013  10017  10043  12799  12802
WAITTM  033552       9880*  9903#  9904*  9952   9962   12802
WAIT.P  033554       9904#
WAIT.T  034016       9993#  11325
WAT   = 104413       2778   2990   3145   3269   3377   3513   3540   3668   3761   3786   3988   4097   4267
                     4369   4563   4707   4759   4962   5111   5339   5483   5600   5668   5812   5971   6148
                     6414   6615   6773   6845   7018   7169   7356   7391   7477   7700   7779   7932   8103
                     8297   8625   8759   8860   8949   9031   9147   9261   9355   9409   9488   11325#
WC      002254       1891#  2824   3033   3189   3431   3592   3843   4018   4145   4296   4459   4620   4815
                     5017   5166   5394   5535   5720   5857   6246   6492   6644   6933   7054   7224   7541
                     7843   7987   8162   8366   8530   10257  11002  12834
WCE   = 040000       1641#
WCF   = 000040       1697#
WCU   = 000001       1770#
WLE   = 004000       1703#
WRCHDT  002340       1929#
WRCHEK  002336       1928#
WRFROM  002370       1946#  2250   2882   2894   2928   2948   3006   3063   3068   3208   3285   3295   3303
                     3313   3334   3417   3550   3558   3609   3796   3804   3861   3921   3946   4004   4044
                     4163   4199   4224   4283   4316   4328   4386   4407   4485   4495   4501   4641   4877
                     4887   4920   5002   5037   5185   5231   5242   5252   5276   5295   5431   5443   5913
                     5923   5939   5999   6011   6022   6043   6055   6065   6088   6107   6159   6285   6297
                     6309   6317   6511   6541   6549   6673   6718   6728   6741   6792   6804   6862   6876
                     6958   7074   7109   7242   7301   7311   7323   7403   7409   7563   7626   7636   7644
                     7654   7667   7712   7718   7730   7829   7872   7878   8002   8039   8177   8266   8268
                     8408   8574   8590   8708   8724   8967   8978   8992   9090   9101   9178   9293   9304
                     9318   9428   9437   9502
WRIDAT  002342       1930#  4335   4360   5450   5475   6811   6837   7737   7763
WRIFOR  002344       1931#  2935   2955   2981   3341   3366   3387   3953   3979   4231   4258   4927   4952
                     4972   5283   5302   5331   5946   5963   6095   6114   6140   6748   6765   7330   7347
```

```
                    7674    7691    8427    8597    8614    8731    8748    8999    9016    9325    9342
WRL    = 004000     1685#
WRU    = 000400     1778#
WSU    = 000004     1772#
XE2      007524     2324    2354#
$AUTOB   001134      872#   2127*  10740   10869
$BDADR   001122      867#   7550*   7852*   9818*   9825*  12811   12822   12825   12827
$BDDAT   001126      869#   2275*   2794*   2803*   3393*   3402*   3733*   3747*   4775*   4784*   4978*   4987*   5127*
                    5136*   7948*   7957*   8119*   8128*   9932*   9944*  10017*  10043*  10152*  10328*  12799   12802
                   12805   12807   12817   12819   12822
$BELL    001216      901#  10693   10862   10949   10971
$CHARC   037536    10608*  10618*  10625   10634*  10639#
$CKSWR   040050    10726#  11321
$CMTAG   001100      855#   2035    2036    2044    2050    2051
$CM1   = 000006      887#    888#    889#    890#    891#    892#    893#
$CM2   = 000014      887#    888#    889#    890#    891#    892#    893#
$CM3   = 000006      885#    887
$CM4   = 000006      893#    894#    895#    896#    897#    898#    899#
$CNTLC   040621    10681   10762   10862#
$CNTLG   040633    10747   10864#
$CNTLU   040626    10772   10863#
$CRLF    001223      903#   2388    2416   10209   10607   10642   10783   10862   10923   10957   10971   11152   11176
                   11181
$DBLK    037312    10529   10563   10571#
$DOAGN   033114     9636    9645    9651#
$DTBL    037302    10532   10567#
$ENDAD   033104      789    9647#
$ENDCT   033052     9638#
$ENDMG   033123     9595    9640    9655#
$ENULL   033120     9598    9643    9654#
$EOP     033016     2218    2349    2427    9603    9628#
$EOPCT   033044     9635#   9639
$ERFLG   001103      858#  10447   10476   10478   10484*  10505   10944*  10971
$ERMAX   001115      864#   2052*  10478   10500*  10505
$ERROR   041022     2044    2234   10937#
$ERRPC   001116      865#  10951*  10952*  10953   10971   11158   12799   12802   12805   12807   12809   12811   12814
                   12817   12819   12822   12825   12827   12829   12831   12834
$ERRTB   001226      920#  11166
$ERRTY   041172    10956   10989#
$ERTTL   001112      862#   9586    9588*  10950*  10971
$ESCAP   001214      900#   2051*  10499*  10966   10968   10971
$FILLC   001156      883#  10611   10642
$FILLS   001155      882#  10642
$GDADR   001120      866#
$GDDAT   001124      868#   2276*   2790*   2799*   3389*   3398*   3734*   3748*   4771*   4780*   4974*   4983*   5123*
                    5132*   7944*   7953*   8115*   8124*  10151*  10327*  12805   12817   12819   12822
$GET42   033074     9644#
$GTSWR   040140    10748#  11319
$HD    = 000000      613
$HIOCT   041020    10911*  10922#
$ICNT    001104      859#  10491*  10492   10494*  10504
$ILLUP   042776    11330   11346   11365#
$INTAG   001135      873#  10745*  10764   10784   10869
$IIEMB   001114      863#  10953*  10971   11155
$LF      001224      904#  10642   10853   10862   10923   10971
$LPADR   001106      860#   2053*   2457*  10224*  10235   10482*  10497*  10502   10504
```

```
$LPERR  001110          861#   2054*  8233*  9084*  9376*  10207  10234*  10482  10498*  10504  10965
$MAIL = ****** U       2071   2123   10497  10595  10959
$MNEW   040651        10751  10867#
$MSWR   040640        10748  10865#
$MXCNT  037074        10495  10504#
$NULL   001154          881#  10613  10642
$NWTST= 000001         2180#  2182   2259#  2261   2282#  2284   2442#  2444   2587#  2589   2629#  2631   2676#
                       2678   2719#  2721   2850#  2852   3223#  3225   3622#  3624   3898#  3900   4173#  4175
                       4664#  4666   4832#  4834   5201#  5203   5410#  5412   5551#  5553   5739#  5741   5876#
                       5878   6693#  6695   7087#  7089   7260#  7262   7578#  7580   8019#  8021   8189#  8191
                       8388#  8390   8545#  8547   8672#  8674   8891#  8893   9209#  9211   9563#  9565
$OCNT   042542        11237* 11266* 11279#
$OMODE  042544        11232* 11236* 11241  11244* 11255* 11281#
$OVER   037060        10459  10475  10483  10493  10501#
$PASS   001100          856#  9594*  9596   9632*  9633*  9641   9654   10489  10505
$POWER  043004        11361  11368#
$PWRAD  042772        11363#
$PWRDN  042632         2048  11330#  11358
$PWRMG  042766        11361#
$PWRUP  042704        11340  11346#
$QUES   001222          902#  10642  10802  10846  10862  10920  10923  10971
$RDCHR  040412        10815#  11322
$RDDEC= ****** U       11325
$RDLIN  040502        10838#  11323
$RDOCT  040662        10884#  11324
$RDSZ = 000011        10831#
$REGAD  001160          885#
$REGO   001162          887#
$REG1   001164          888#
$REG2   001166          889#
$REG3   001170          890#
$REG4   001172          891#
$REG5   001174          892#
$RTNAD  033116         9653#
$R2A  = ****** U       11325
$SAVRE= ****** U       11325
$SAVR6  043002        11339* 11347  11348* 11349* 11367#
$SCOPE  036624         2042   10456#
$SETUP= 000117         2033#  2041   2042   2044   2046   2048   2050   2051   2053   2120   9630   10457  10685
                       10690  10691  10721  10869  10938  10962  10970
$SS1  = 000000         2072#
$STUP = 177777         2033#
$SVLAD  037032        10467  10496#
$SVPC = 000200          787#   792
$SWR  = 167770          603#   613    651    652    653    654    655    656    657    658    899    900    901
                       2050   2051   2053   2054   2187   2266   2296   2456   2596   2638   2683   2728   2864
                       3241   3637   3910   4188   4675   4866   5211   5420   5564   5749   5902   6706   7098
                       7280   7602   8028   8200   8398   8559   8693   8911   9225   9574   9625   9631   9646
                       9652   9654   10448  10449  10450  10451  10452  10458  10470  10472  10473  10476  10477
                       10478  10485  10486  10487  10498  10501  10504  10929  10930  10931  10932  10933  10947
                       10954  10959  10963  10971  11364
$SWRMK= 000000          658    659   10452  10453  10474
$TIMES  001212          899#  2050*  2187*  2266*  2296*  2456*  9574*  9631*  10485*  10492  10495*  10504
$TKB    001146          878#  10645  10666  10677  10702  10730  10757
$TKCNT  037542        10646#  10661* 10691  10708* 10822  10824*
$TKINT  037562         2080   2302   2732   10359  10661#  10682  10743
```

CROSS REFERENCE TABLE -- USER SYMBOLS


```
$TKQEN= 037561          10650#  10716   10827
$TKQIN  037544          10647#  10662*  10663   10714*  10715*  10716   10718*
$TKQOU  037546          10648#  10663*  10825   10826*  10827   10829*
$TKQSR  037550          10649#  10662   10718   10829
$TKS    001144            877#  10645   10667*  10698*  10700   10706*  10728   10744*  10754   10766*  10786*
$TKSRV  037632          10664   10677#
$TMP0   001176            893#   2364*  2365*
$TMP1   001200            894#   2204*  2656*  2657*  2701*  2702*  8215*  8218    8220*  8224    8228    8229*  8924*
                         9059*   9074*  9202*  9277*  9280*  9368*  9377*  9516    9518*  9527    9529    9533*  9534
                         9535    9538   9547   9550   9554*  12809
$TMP2   001202            895#
$TMP3   001204            896#
$TMP4   001206            897#
$TMP5   001210            898#
$TN   = 000041            603#    613   2163   2174   2180   2187#  2188    2259    2266#  2272    2282    2296#  2298
                         2436    2442   2456#  2459   2587   2596#  2604    2629    2638#  2639    2662    2668   2676
                         2683#   2684   2706   2719   2728#  2735   2850    2864#   3223   3241#   3622    3637#  3898
                         3910#   4173   4188#  4664   4675#  4832   4866#   5201    5211#  5410    5420#   5551   5564#
                         5739    5749#  5876   5902#  6693   6706#  7087    7098#   7260   7280#   7575    7578   7602#
                         8015    8019   8028#  8189   8200#  8210   8388    8398#   8545   8559#   8672    8693#  8777
                         8891    8911#  9204   9209   9225#  9563   9574#
$TPB    001152            880#  10631*  10642
$TPFLG  001157            884#  10589   10642
$TPS    001150            879#  10629   10642
$TRAP   042546           2046   11290#
$TRAP2  042570          11301#  11312
$TRP  = 000014          11305#  11314#  11315#  11316#  11317#  11318#  11319   11320#  11321   11322#  11323#  11324#  11325#
                        11326#
$TRPAD  042602          11295   11312#
$TSTNM  001102            857#   9589*  9630*  10447   10474   10496*  10501   10505   10946   10971
$TTYIN  040610          10839   10840   10857  10861#
$TYPBN= ****** U        11318
$TYPDS  037076          10517#  11317
$TYPE   037322          10589#  11305   11313
$TYPEC  037472          10610   10617   10624  10629#  10630   10788
$TYPEX  037540          10635   10637   10640#
$TYPOC  042344          11235#  11314
$TYPON  042360          11234   11237#  11316
$TYPOS  042320          11230#  11315
$XTSTR  036636          10461#
$$GET4= 000000           9646#
$OFILL  042543          11231*  11235*  11245  11280#
$40CAT= ****** U        10458   10956
.     = 062662            775#    779#   787    788#   790#   792#   796#   800#   847#   854#   905    1945#  1946#
                         1947#   1964#  1980#  2039   2053   2054   2099#  2103#  2107#  2227#  2233#  2309#  2335#
                         2339#   2347#  2400#  2406#  2412#  2495#  2525#  2564#  2570#  2576#  2614#  2618#  2624#
                         9579#   9585#  9654   9658   10213#  10221#  10228#  10375#  10387#  10393#  10399#  10411#  10432#
                        10504   10505   10571#  10642  10645   10649#  10650   10651#  10861#  10862   10869   10923   10971
                        11001#  11009#  11073#  11081#  11089#  11097#  11105#  11113#  11137#  11145#  11203#  11342  11366
                        12797#  12880#
```

```
ALLREG   647#  10996  11005  11013  11021  11029  11037  11045  11053  11061  11069  11077  11085  11093  11101
        11109  11117  11125  11133  11141
CHANGR   647#   2808   3008   3015   3164   3171   4105   4114   4123   4135   4387   4409   4418   4433   4437
         4444   4581   4588   4602   4609   4791   4798   4805   5362   5371   5378   5506   5517   5524   5691
         5702   5709   5820   5830   5839   5846   6166   6175   6189   6200   6432   6441   6480   6879   6889
         6896   6910   6914   7178   7187   7201   7208   7485   7492   7504   7513   7517   7787   7794   7801
         7810   7814   8305   8312   8321   8327   8338   8473   8482   8496   8503   8507
CHECKD   647#   2970   3125   3249   3353   3500   3646   3708   3966   4075   4245   4347   4541   4684   4737
         4939   5088   5318   5462   5574   5648   5791   5950   6127   6393   6594   6752   6824   6997   7148
         7334   7365   7448   7678   7750   7909   8075   8289   8423   8601   8735   8837   8928   9003   9126
         9232   9329   9462
CHECKV   647#   2742
CHKCNT   647#
CKCNTV   647#
CLEARA   647#   2905   3066   3293   3311   3319   3556   3802   3931   4042   4209   4314   4493   4499   4507
         4885   4905   5054   5250   5429   5614   5758   5921   6020   6063   6307   6315   6323   6547   6555
         6726   6790   6956   6964   7107   7115   7309   7401   7407   7415   7634   7652   7710   7716   7870
         7876   8037   8044   8975   9077   9098   9301   9417   9434
CMPBLK   647#   3205   3606   3858   4160   4638   5033   5182   6508   6670   7071   7239   7560   7999   8174
         9175   9500
CMREGI   647#   2820   3029   3185   3427   3588   3839   4014   4141   4292   4455   4616   4811   5013   5162
         5390   5531   5716   5853   6242   6488   6640   6929   7050   7220   7537   7839   7983   8158   8362
         8526
COMMEN     1#    616    772#
DATACO   647#   2921   2940   3081   3099   3327   3462   3681   3939   4050   4217   4322   4514   4913   5062
         5269   5287   5437   5622   5766   5932   6081   6099   6345   6363   6569   6734   6798   6972   7123
         7316   7422   7660   7724   7884   8051   8582   8716   8820   8984   9107   9310   9443
DISREG   647#   6921   7041   7212
DUM      647#   2745
ENDCOM     1#    625    772#
ERROR    666#   2205   2278   2660   2671   2711   2796   2805   2830   3039   3195   3215   3395   3404   3437
        3597   3616   3737   3751   3849   3868   4023   4150   4169   4301   4464   4626   4647   4777   4786
        4821   4980   4989   5023   5043   5129   5138   5172   5192   5400   5541   5726   5863   6252   6498
        6518   6650   6680   6939   7060   7081   7230   7249   7551   7570   7853   7950   7959   7993   8009
        8121   8130   8168   8184   8372   8536   8640   8774   8874   8884   9049   9165   9186   9509   9819
        9826   9935   9937   9947   9956   9965  10021  10025  10044
ESCAPE     1#    772#
FIHEAD   647#   2880   2891   3061   3283   3301   3548   3794   3919   4197   4483   4875   4897   5229   5239
         5911   5997   6008   6041   6052   6283   6294   6539   6716   7299   7624   7642   8572   8706   8964
         9087   9290   9425
FILLBL   647#
FLSVRE   647#   2998   3003   3022   3154   3159   3178   3409   3414   3419   3564   3569   3574   3579   3810
         3815   3820   3825   3830   3996   4001   4006   4130   4275   4280   4285   4398   4404   4428   4571
         4576   4597   4994   4999   5004   5143   5148   5153   6156   6161   6184   6213   6218   6224   6230
         6422   6427   6452   6465   6471   6623   6628   6633   6867   6873   6905   7026   7031   7036   7196
         7499   7524   7529   7821   7826   7831   7964   7970   7975   8135   8140   8145   8150   8348   8354
         8491
GETPRI     1#    772#
GETSWR     1#    603#   772#   2120
GOO      647#   2766   2980   3135   3261   3365   3504   3660   3725   3978   4087   4257   4359   4553   4696
         4749   4951   5100   5330   5474   5591   5659   5803   5962   6139   6405   6606   6764   6836   7009
         7160   7346   7382   7460   7690   7762   7921   8088   8613   8747   8847   8940   9015   9138   9252
         9341   9400   9479
LOAD     647#
MSG     2180#  2182   2258#  2261   2281#  2284   2441#  2444   2587#  2589   2628#  2631   2675#  2678   2718#
        2721   2849#  2852   3222#  3225   3621#  3624   3897#  3900   4173#  4175   4663#  4666   4832#  4834
```

```
        5200#   5203    5409#   5412    5550#   5553    5738#   5741    5875#   5878    6692#   6695    7087#   7089    7259#
        7262    7578#   7580    8018#   8021    8188#   8191    8387#   8390    8544#   8547    8671#   8674    8890#   8893
        9208#   9211    9563#   9565
MULT       1#    772#
NEWTST     1#    772#  2180    2259    2282    2442    2587    2629    2676    2719    2850    3223    3622    3898    4173
        4664    4832    5201    5410    5551    5739    5876    6693    7087    7260    7578    8019    8189    8388    8545
        8672    8891    9209    9563
OFFST    647#
POP        1#    772#  7552    7854    9684    9712    9737    9854    9967   10049   10077   10109   10164   10262   10338
       10558   10912   11351   11352
PUSH       1#    772#  7547    7849    9673    9702    9730    9844    9907    9993   10064   10094   10132   10252   10310
       10517   10886   11332   11338
REPORT     1#    772#
RFORGC   647#
RHCLEA   647#   2461    2643    2688    2737    2866    3053    3243    3276    3456    3639    3676    3912    4036    4190
        4475    4677    4714    4868    5048    5213    5422    5566    5607    5751    5904    5977    6269    6330    6525
        6562    6708    6780    6949    7100    7282    7293    7604    7615    7862    8030    8202    8236    8400    8561
        8647    8695    8784    8812    8913    8956    9067    9227    9283    9386
SAVE     647#  10939
SAVTST   647#   2188    2298    2596    2639    2684
SCH      647#   3713    4721
SCOPE    667#   2186    2265    2295    2455    2595    2637    2682    2727    2863    3240    3636    3909    4187    4674
        4865    5210    5419    5563    5748    5901    6705    7097    7279    7601    8027    8199    8397    8558    8692
        8910    9224    9573    9629
SEEKCO   647#   3656    5586    7370
SETPRI     1#    772#  10818
SETTRA 11305#  11314   11315   11316   11317   11319   11321   11322   11323   11324   11325
SETUP      1#    772#  2034
SKIP       1#    647#   772#   2163    2174    2272    2436    2604    2662    2668    2706    9204
SLASH      1#    772#
SPACE    772#
SREGIS   647#   2752    2963    3119    3346    3481    3700    3959    4068    4238    4340    4534    4729    4932    5081
        5311    5455    5641    5784    6120    6386    6588    6817    6990    7141    7441    7743    7902    8068    8281
        8434
STARS      1#    639     645    772#    785     850     905    1606    1608    1653    1655    2134    2156    2180    2185
        2259    2264    2282    2294    2374    2394    2442    2454    2507    2518    2544    2577    2587    2594    2629
        2636    2676    2681    2719    2726    2850    2862    2874    2878    2916    2919    3076    3079    3223    3239
        3622    3635    3877    3891    3898    3908    4173    4186    4664    4673    4832    4864    5201    5209    5223
        5227    5262    5266    5388    5410    5418    5551    5562    5739    5747    5876    5900    5931    5988    5991
        5995    6035    6039    6075    6079    6207    6211    6268    6277    6281    6340    6344    6460    6464    6536
        6693    6704    7087    7096    7260    7278    7578    7600    8019    8026    8189    8198    8388    8396    8545
        8557    8672    8691    8891    8909    9209    9223    9271    9275    9380    9521    9525    9541    9545    9562
        9563    9572    9622   10427   10436   10444   10507   10574   10644   10721   10736   10807   10831   10872   10925
       11207   11284   11328   11344
STARTT   647#   2459    2735    2864    3241    3637    3910    4188    4675    4866    5211    5420    5564    5749    5902
        6706    7098    7280    7602    8028    8200    8398    8559    8693    8911    9225
SWRSU      1#    772#  2055#
TJUMP    647#   7575    8015    8210    8777
TRMTRP 11305#
TSCLR2   647#
TSCLR5   647#
TTSTNO   647#   2460    2736    2865    3242    3638    3911    4189    4676    4867    5212    5421    5565    5750    5903
        6707    7099    7281    7603    8029    8201    8399    8560    8694    8912    9226
TYPBIN     1#    772#
TYPDEC     1#    772#  9641
TYPNAM     1#    772#
```

```
TYPNUM      1#    772#
TYPOCS      1#    772#
TYPOCT      1#    772#  2532  2538 10749
TYPTXT      1#    772#  2086  2091  2096  2100  2104  2109  2113  2166  2209  2224  2230  2306  2331
         2336    2340  2344  2397  2403  2409  2487  2492  2522  2528  2534  2560  2566  2572  2607
         2611    2615  2621  9576  9582 10197 10203 10210 10214 10218 10225 10229 10348 10355 10371
        10378   10384 10390 10396 10400 10404 10408 10414 10419 10428 10998 11006 11014 11022 11030
        11038   11046 11054 11062 11070 11078 11086 11094 11102 11110 11118 11126 11134 11142
VECSET    647#  2759  2974  3129  3254  3358  3526  3650  3773  3971  4080  4250  4352  4546  4689
         4742    4944  5093  5323  5467  5579  5653  5796  5955  6132  6398  6599  6757  6829  7002
         7153    7339  7375  7453  7683  7755  7914  8080  8606  8740  8841  8933  9008  9131  9245
         9334    9393  9467
WT        647#  2777  2989  3144  3268  3376  3512  3539  3667  3760  3785  3987  4096  4266  4368
         4562    4706  4758  4961  5110  5338  5482  5599  5667  5811  5970  6147  6413  6614  6772
         6844    7017  7168  7355  7390  7476  7699  7778  7931  8102  8296  8624  8758  8859  8948
         9030    9146  9260  9354  9408  9487
WTT       647#  2783  2995  3150  3274  3382  3518  3545  3673  3766  3791  3993  4102  4272  4374
         4568    4712  4764  4967  5116  5344  5488  5605  5673  5817  5976  6153  6419  6620  6778
         6850    7023  7174  7361  7396  7482  7705  7784  7937  8108  8302  8630  8764  8865  8954
         9036    9152  9266  9360  9414  9493
$$CMRE    848#   887   888   889   890   891   892
$$CMTM    848#   893   894   895   896   897   898
$$ESCA      1#   772#
$$NEWT      1#   772#  2180  2259  2282  2442  2587  2629  2676  2719  2850  3223  3622  3898  4173
         4664    4832  5201  5410  5551  5739  5876  6693  7087  7260  7578  8019  8189  8388  8545
         8672    8891  9209  9563
$$SET   11305#  11314 11315 11316 11317 11319 11321 11322 11323 11324 11325
$$SKIP      1#   772#
.EQUAT      1#   603#   662
.HEADE      1#   603#
.KT11       1#   603#   811
.SETUP      1#   603#  2032
.SWRHI      1#   603#   647
.SWRLO    603#   659#
.$ACT1      1#   603#   783
.$APTB      1#
.$APTH      1#
.$APTY      1#
.$ASTA      1#
.$CATC      1#   603#   773
.$CMTA      1#   603#   848
.$DB2D      1#
.$DB2O      1#
.$DIV       1#
.$EOP       1#   603#  9620
.$ERRO      1#   603#  10923
.$ERRT      1#   603#
.$MULT      1#
.$POWE      1#   603#  11326
.$RAND      1#
.$RDDE      1#
.$RDOC      1#   603#  10870
.$READ      1#   603#  10642
.$R2AZ      1#
.$SAVE      1#
.$SB2D      1#
```

```
.$SB20      1#
.$SCOP      1#    603#  10442
.$SIZE      1#
.$SUPR      1#
.$TRAP      1#    603#  11282
.$TYPB      1#
.$TYPD      1#    603#  10505
.$TYPE      1#    603#  10572
.$TYPO      1#    603#  11205
.$40CA      1#
.1170       1#


. ABS.  062662     000


 ERRORS DETECTED:  0

 DSKZ:CZRJJD.BIN,DSKZ:CZRJJD.LST/CRF/SOL/NL:TOC:MD:MC:CND/LI:ME=CZRJJD.SML,CZRJJD.P11
 RUN-TIME: 103 142 7 SECONDS
 RUN-TIME RATIO: 562/254=2.2
 CORE USED:  37K  (73 PAGES)
```